Monitoring and Fault Diagnosis of Fermentation Processes

Gregersen, Lars

Publication date:
2004

Document Version
Publisher's PDF, also known as Version of record

Monitoring and Fault Diagnosis of Fermentation Processes

Lars Gregersen

June, 2004

Novo Nordisk A/S and Department of Chemical Engineering, Technical University of Denmark.

ISBN 87-91435-09-9
Printed by Book Partner, Nørhaven Digital, Copenhagen, Denmark

Preface

This thesis is submitted as partial fulfillment of the requirements for the Ph.D. degree at the Technical University of Denmark. The Ph.D. carried out has been an industrial Ph.D. funded by The Academy of the Technical Sciences and the pharmaceutical company Novo Nordisk where a great deal of the work has been carried out.

I would like to thank for the valuable help and guidance I have received from my two supervisors Professor Sten Bay Jørgensen and Ph.D. Maria Yolanda Andersen.

Thanks goes the people at Novo Nordisk whom I have worked with and without their help I would not have had any process data to work with. A special thank goes to Uffe Clausen, Mikael Bundgaard, Mikael Dahm-Hansen and Anders Gram.

I would like to thank the Ph.D. students and master's students I have worked with at Department of Chemical Engineering. We have had many interesting discussions regarding batch modelling, control and optimisation that I have learned a lot from. In no particular order: Hassan Yazdi, Bodil Recke, Britta R. Andersen, Frede Lei, Niels Rode Kristensen, Kurt Creutzburg, Mario Eden, René Skotte and Dennis Bonné.

For assistance in the areas of numerical analysis and chemometrics I would like to thank Per Christian Hansen and Agnar Höskuldsson.

Finally, I would like to thank Susan and my parents for helping me through difficult times while writing this thesis.

Kgs. Lyngby, June 2004

Lars Gregersen

Abstract

This Ph.D. thesis is the result of a business Ph.D. study carried out with Fermentation Pilot Plant, Novo Nordisk A/S and Department of Chemical Engineering, The Technical University of Denmark.

The thesis focuses on how to model biochemical batch processes using available process data. It is assumed that the process is so complex that first principles modelling of the process it time consuming and therefore too expensive to be viable. This type of modelling is therefore not considered in this thesis. Instead the modelling will be based on process data.

This thesis falls in two parts. The first part is about data mining and the use of process chemometrics for faults diagnosis. The second part is about the development of an LTI model for batch input-output modelling

Chapter 1 introduces the assumptions and goals of this thesis and gives the motivation for this work. A short introduction to data driven modelling is given. This introduction is extended in chapter 2.

Chapter 2 Focuses on data mining and the aspects of data quality which must be constantly monitored in order to obtain sufficiently informative and reliable models through data driven modelling. Attention must be applied to all parts in the data collection pipeline such that the data collection is as informative as the application requires. Pre-treatment and validation of data before they are used for any detailed modelling is also importent. Here chemometric tools can be of great value.

Chapter 3 gives a treatment on chemometric modelling tools. Chemometrics is used as data mining tools and for building regression models. It is shown how principal component analysis (PCA) may be used to extract hidden knowledge in the form of latent variables by an eigenvector decomposition of the covariance matrix. Usually only a few latent variables will suffice to accurately describe the process behaviour. Principal component regression (PCR) and projection to latent structures (PLS) are chemometrical work horses that have more robust properties than the ordinary least squares solution commonly used for linear regression type of problems.

Methods for fault diagnosis are described. These methods use chemometric decompositions to extract relevant information from the data in a form that makes it suitable for graphical display. These graphical displays may be used by the process operators to assess when a process is running under normal operating conditions or when an abnormal event has occurred. Fault diagnosis consists of three sequential steps: Detection, isolation and identification. The detection task is to monitor the process and detect *if* and *when* a fault has

occurred. Once a fault has been detected it is important to determine what variables contribute to the fault. This is done in the isolation step. Elimination of a fault requires detailed process knowledge. This knowledge is not available in the chemometrical models developed and it is therefore still a manual operator's task to handle fault using fault identification.

Chapter 4 Offers an example of process chemometrics for faults diagnosis. The main example is an industrial fed-batch fermentation process for which some process data are available. The use of PLS models is illustrated. Process data pre-treatment and screening is carried out and PLS models are identified that are used in the identification of faults.

It is shown that the described process chemometric methods indeed can be applied to fault diagnosis even though the process data used for the modelling were standard process data.

Chapter 5 presents a new model structure for dynamic batch input/output modelling. Methods are also given for the identification of the parameters in the model. The model type is called a stacked state space model, which is short for finite-horizon, time-shifted, stacked, linear state space models. The model structure is similar to models obtained when linearising first principles models, but does only require input/output data. The models become large with many parameters to be estimated. By the usage of regularisation methods the bias contra variance tradeoff is taken care of and the methodologies given provide reliable model even for a very large number of parameters. Examples in this chapter are given with simulated data only.

Chapter 6 shows the relationship between the chemometric methods and the dynamic models developed. These two areas share many aspects regarding parameter estimation, numerical stability and interpretability of the models.

It is shown that the models for faults diagnosis are similar in design when compared to the dynamic models and that the estimation routines applied are similar with only small differences. The main difference between the chemometric model and the dynamic model is that it is possible to include causality constraints in the dynamic models. This is not directly possible with the chemometric models which represent the correlation between variables. However, a scheme is shown that will allow the combination of the chemometric models with the dynamic batch models to obtain a fault diagnosis system that would be able to give more detailed detection and isolation capabilities and possibly some insight into the fault identification problem.

Chapter 7 demonstrates how stacked state space models can be applied to process optimisation and control using the readily available methods developed in the field of process control.

The model structure defined is directly suitable for the inclusion in batch optimisation and batch control formulations. Since the developed model structure is linear the resulting problems become QP problems which are easily solved.

A control structure is given for a proportional controller that utilises the information in the model to obtain good tracking performance of a reference

trajectory. An example is given where this controller is used to control a batch and where subsequent batches when included in the modelling may improve the tracking capabilities.

Chapter 8 ends this thesis with the main conclusions and suggestions for future work.

Resumé (in Danish)

Denne afhandling er et resultat af et erhvervsforskerprojekt udført i samarbejde mellem Institut for Kemiteknik, Danmarks Tekniske universitet og Gærings Pilot Plant, Novo Nordisk A/S.

Denne afhandling fokuserer på hvordan biokemiske processer kan modelleres under anvendelse af normalt tilgængelige processmålinger. Det antages at de biologiske processer er så komplekse, at modeller baseret på fundementale principper og hastighedsudtryk ikke vil kunne udvikles tilstrækkeligt hurtigt til at være praktisk anvendelige. Modellering vil derfor baseres på procesdata og statistiske principper.

Afhandlingen falder i to dele. Den første del omhandler data mining og anvendelsen af proceskemometri til fejldiagnose. Den anden del omhandler udviklingen af en linear tidsinvariant model til batchmodellering.

Kapitel 1 præsenterer motivationen for det udførte arbejde. Mål og forudsætning defineres. En kort præsentation af databaseret modellering gives. Denne præsentation uddybes i kapitel 2.

Kapitel 2 giver en gennemgang af data mining og giver retningslinier for optagelse af informative data til anvendelse til modellering. Der må generelt anvendes meget tid på screening og validering af data før de anvendes til egentlig databaseret modellering.

Data mining kræver computerprogrammer som kan håndtere store datamængder og som kan foretage den nødvendige analyse af data. Disse værktøjer må udvikles så de tager specielt hensyn til batch processer.

Kapitel 3 omhandler kemometriske metoder. Disse modelleringsmetoder anvendes til process analyse og fejldiagnose. Principal komponent analyse (Principal Components Analysis, PCA) anvendes til data- og processanalyse. Regressions modeller såsom Principal komponent regression (Principal Components Regression, PCR) og projektion på latente strukturer (Projection to Latent Structures, PLS) anvendes i stedet for ordinær linear regression.

Metoder til fejldiagnose under brug af kemometriske modeller er vist. Fejldiagnosen har til formål at detektere fejl hvis og når de opstår. Fejlisolering har til formål at identificere hvilke målte variable der signalerer en opstået fejl. Fejlidentifikation anvendes når fejlen skal afhjælpes og den kausale struktur der ligger til grund for fejlen skal klarlægges.

Kapitel 4 giver et praktisk eksempel på anvendelsen af processkemometri på industrielle data fra en fed-batch gæringsprocess. PCA og PLS modellering er anvendt til at finde strukturer i data og til fejldiagnose af opståede fejl i nogle af de analyserede batche.

De vises at det er muligt at konstruere brugbare modeller baseret på standard procesdata.

Kapitel 5 præsenterer en ny lineær tidsinvarient model struktur for dynamisk batch input/output modellering. Der vises pålidelige metoder baseret på regularisering til at estimere parametrene i modellerne. Eksempler i dette kapitel er baserede på simulerede data.

Den udviklede modelstruktur er direkte anvendelig til inkludering i batch optimering og batch kontrol problemer. Da model strukturen er linear resulterer dette i QP problemer, der let løses. En proportional regulator til løsning af batch control problemet er defineret og præsenteret ved et simuleringeksempel.

Kapitel 6 viser sammenhængen mellem de behandlede kemometriske modeller og de dynamiske batch modeller udviklet i dette arbejde. Det er vist at der er en klar sammenhæng mellem de to model typer og at den dynamiske model kan konstrueres så den giver en kausal model i modsætning til de kemometriske modeller der modellerer korrelationerne mellem data.

En algoritme er givet for process overvågning hvor den dynamiske model inddrages sammen med de kemometriske modeller til at detektere fejl under hensyntagen til separationen af data i input og outputs.

Kapitel 7 viser hvor de udviklede batch arx modeller kan bruges til proceskontrol of procesoptimering ved brug at den eksisterende teori fra proceskontrolområdet herunder model prædiktion kontrol (MPC).

Kapitel 8 giver en kort opsummering af resultaterne i denne afhandling og præsenterer forslag for fremtidigt arbejde.

Contents

Preface iii

Abstract v

Resumé (in Danish) ix

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Industrial constraints . . . 2
- 1.3 The Control Hierarchy . . . 2
  - 1.3.1 Data Quality . . . 3
  - 1.3.2 Fault Diagnosis . . . 4
  - 1.3.3 Fault recovery . . . 4
- 1.4 Modelling batch processes . . . 4
- 1.5 Literature review . . . 8
- 1.6 Scope and Outline of the Thesis . . . 9
  - 1.6.1 Hypothesis . . . 9
  - 1.6.2 Achievements . . . 9
  - 1.6.3 Scope of the thesis . . . 10
  - 1.6.4 Outline . . . 10

2 Data Driven Modelling 13
- 2.1 An introduction to process data . . . 14
- 2.2 Measurements . . . 15
  - 2.2.1 Measurement devices and soft sensors. . . . 16
  - 2.2.2 Tools for data storage, retrieval and analysis. . . . 18
- 2.3 Data preparation . . . 20
- 2.4 Modelling . . . 23
  - 2.4.1 Data analysis . . . 23
  - 2.4.2 Modelling tools. . . . 24
- 2.5 Applications . . . 25
  - 2.5.1 Process Monitoring . . . 25
  - 2.5.2 Process Control . . . 25
- 2.6 Discussion . . . 26
- 2.7 Conclusion . . . 27

3 Process Chemometrics 29
- 3.1 Principal Component Analysis . . . 31
  - 3.1.1 Eigenvectors and Eigenvalues . . . 32
  - 3.1.2 Interpretation of the Principal Components . . . 33

3.1.3 Scaling . . . 34
3.2 Number of Principal Components . . . 35
3.2.1 Covariance matrix of less than full rank . . . 35
3.2.2 Eigenvalues Almost Equal . . . 36
3.2.3 Statistical Tests. . . . 36
3.2.4 Explanation of Variance . . . 37
3.2.5 Inspection of Eigenvalues . . . 37
3.2.6 Cross-validation. . . . 38
3.3 Calculating the PCA in detail . . . 39
3.3.1 Eigenvalue Method . . . 39
3.3.2 NIPALS . . . 40
3.3.3 SVD . . . 42
3.4 Linear Regression . . . 43
3.4.1 Introduction . . . 43
3.4.2 Multivariate Linear Regression . . . 44
3.4.3 Ordinary Least Squares . . . 44
3.4.4 Maximum Likelihood Estimation . . . 47
3.4.5 Ridge Regression . . . 49
3.4.6 Principal Component Regression . . . 49
3.4.7 Regression . . . 50
3.5 Projection to Latent Structures . . . 50
3.5.1 PLS1: One $\mathbf{y}$-variable . . . 51
3.5.2 PLS2: More than one $\mathbf{y}$-variable . . . 52
3.5.3 Regression . . . 53
3.6 Unfolding . . . 55
3.6.1 Three-Way Principal Component Analysis . . . 56
3.6.2 Multi-Way PCA . . . 56
3.7 Fault Diagnosis . . . 57
3.7.1 PCA . . . 58
3.7.2 Q-Statistic . . . 59
3.7.3 MPCA . . . 61
3.7.4 Interpretation of the Principal Components . . . 61
3.7.5 T^2 statistic . . . 63
3.7.6 On-line Estimation of t-scores . . . 64
3.7.7 Kernel density estimate of confidence area . . . 64
3.7.8 Squared Prediction Error . . . 65
3.7.9 T_f^2 statistic . . . 66
3.8 PLS—Fault Diagnosis . . . 67
3.8.1 Fault Detection and Isolation . . . 68
3.8.2 Squared Prediction Error . . . 69
3.8.3 T_f^2 statistic . . . 70
3.9 Summary . . . 70
3.9.1 Future research . . . 71
List of Symbols . . . 71

4 Supervision of Fed-Batch Fermentations 73
4.1 Introduction . . . 73
4.2 Process Description . . . 74
4.3 Data Analysis . . . 75
4.3.1 Fault Diagnosis . . . 79
4.4 Experimental Results . . . 81
4.4.1 On-line estimation of Final Product Concentration . . . 82
4.4.2 Score plots . . . 84
4.5 Discussion and Conclusion . . . 85

5 Dynamic I/O Modelling for Batch Processes 89
5.1 Dynamic Models . . . 91
5.2 ARX models . . . 91
5.3 Stacked State Space Models for Batch Processes . . . 92
5.4 Parameter Estimation . . . 94
5.4.1 SISO Example . . . 94
5.4.2 Causality Constraints . . . 98
5.4.3 Model Order . . . 103
5.4.4 ARX with constant parameters . . . 106
5.4.5 Regularisation . . . 107
5.5 Simulation . . . 108
5.5.1 Motivation . . . 109
5.5.2 Closed form . . . 109
5.6 MIMO models for batch . . . 111
5.6.1 Matrix Structure . . . 111
5.6.2 Example . . . 112
5.7 Discussion . . . 115
5.8 Conclusion . . . 118
5.8.1 Future Research . . . 118
5.9 SISO Example . . . 120
5.10 MIMO Example . . . 120
List of Symbols . . . 126

6 Dynamic Batch Process Monitoring using Local Linear Models 129
6.1 Introduction . . . 129
6.2 Chemometrics and process control . . . 131
6.3 ARX for continuous processes . . . 132
6.4 Stacked state space models. . . . 133
6.4.1 Estimation . . . 134
6.4.2 Causality Constraints . . . 135
6.4.3 Model Order . . . 136
6.5 Process Chemometrics for Fault Diagnosis . . . 136
6.5.1 Process Chemometrics Methods Combined with Stacked State Space Models. . . . 137
6.5.2 Monitoring . . . 139

6.6 Example . . . 140
6.6.1 Model estimation. . . . 140
6.6.2 Monitoring . . . 141
6.7 Conclusion . . . 143

7 A Conceptual Solution to the Generic Fed-batch Control Problem. 145
7.1 Introduction . . . 145
7.2 Modelling Batch Processes for Control . . . 146
7.3 Stacked state space models . . . 146
7.3.1 Estimation . . . 147
7.3.2 Simulation . . . 149
7.4 Conceptual control of batch processes . . . 150
7.5 Model Predictive Control . . . 152
7.5.1 Optimisation based formulation of the control problem . 152
7.5.2 Analysis of the Batch Control Problem . . . 153
7.6 Batch Control Example . . . 155
7.7 Discussion and Conclusions . . . 160

8 Discussion and conclusions 163
8.1 Fault diagnosis . . . 163
8.2 Stacked State Space Models . . . 165
8.3 Main results . . . 166
8.4 Future work . . . 167

A Linear Algebra 169
A.1 Vector and Matrix Notation . . . 169
A.1.1 Kronecker product . . . 170
A.2 Rank . . . 170
A.3 Eigenvalues and Eigenvectors . . . 170
A.4 Singular Value Decomposition . . . 171
A.4.1 Economy Size SVD . . . 172
A.5 Norm . . . 172
A.5.1 Vector Norm . . . 172
A.5.2 Matrix Norms . . . 172
A.6 Pseudo Inverse . . . 173
A.7 Derivatives . . . 174
A.8 Matrix Exponential Function . . . 174

B Statistical Analysis 177
B.1 Summarising Statistics . . . 177
B.1.1 Sample Mean . . . 177
B.1.2 Sample Variance . . . 178
B.1.3 Sample Correlation . . . 179
B.1.4 Scaling . . . 179

B.2 Distributions 180
B.2.1 Distribution Functions 180
B.2.2 Expectation and Variance 180
B.2.3 The Multinormal Distribution 181
B.2.4 The Wishart Distribution 181
B.2.5 The Hotelling T^2 Distribution 182
B.3 Univariate Statistics 182
B.3.1 Chi-squared Distribution 182
B.3.2 F and Beta Variables 183
B.3.3 t distribution 183
B.4 Bias/Variance dilemma 184

References **187**

1

Introduction

The main goal for this thesis is to investigate data driven methodologies for modelling batch processes. Batch processes are used by the chemical and biochemical industries to produce a wide range of products that must be produced with the specified quantity and quality at the specified time. Methodologies for monitoring batches and assuring consistency between batches are highly desirable and will be further developed in this thesis. Control and optimisation are also very important areas for which a new linear dynamic batch model type is developed.

The emphasis will be on chemometric methods and their related numerical algorithms. These methods are chosen for their ability to handle massive amounts of data for process modelling. The data will be used to form reference process models. It is demonstrated how such reference models whether static or dynamic may be used for off-line and on-line applications of process monitoring and control.

1.1 Motivation

Digital process control system have been used for decades for the control of chemical unit operations. In recent years there has been an increasing effort to store process variables values in large data bases for future reference. These data are used on an *ad hoc* basis for finding process abnormalities or process limitations that need attention in order to reduce faults and variability and to optimise the profitability of the process.

This chapter will start by introducing the industrial problem that is covered in this thesis. Then a description of the fault diagnosis and control problem will follow. This thesis is concerned with batch and fed-batch processes that require special handling of the process data obtained due to the special nature of these process types. A motivation for this work is given. Finally, a detailed outline of the chapters of this thesis is given.

1.2 Industrial constraints

The work described in this thesis is the result of an industrial Ph.D. project. The work has been carried out in cooperation with the Danish company Novo Nordisk A/S and CAPEC at the Department of Chemical Engineering, Technical University of Denmark.

Novo Nordisk A/S is divided into two large divisions. One is Health Care and the other is Enzyme Business. The work has been carried out in the department called Process Technology within the Enzyme Business[1].

The majority of production lines and unit operations at Novo Nordisk A/S are equipped with a process control system that collects data which are stored in data bases. This means that many processes have a large amount of data generated that describe the process operation. It is of interest to control the entire production of enzymes since this would facilitate the control and optimisation of the entire production line. Early in the work it was decided to look mainly at the main bioreactor, simply to assess whether the proposed approaches are feasible. The main bioreactor is the unit operation the is most well equipped with measurement devices. The data collected are stored in data bases available world wide through the intranet of the company.

At Novo Nordisk A/S industrial enzymes are mainly produced by microorganisms that produce enzymes during growth in a fermentation process by digesting a substrate (e.g. glucose). The process is most often carried out as a fed-batch process. This means that the fermentor is initially supplied with a small amount of substrate and microorganisms. During the batch additional substrate will be fed to the fermentor. This feed addition may lead to large changes in concentrations and volume which again affect the dynamic behaviour of the process. The terms batch and fed-batch will be used interchangeably in the thesis.

1.3 The Control Hierarchy

The control hierarchy in a chemical plant consists of the classical control layers (SISO, MIMO, MPC and scheduling layers) as shown in the triangle on the left hand side in figure 1.1. The bottom layer of the control hierarchy consists of single loop controllers where the sample rate usually is high. As one moves up to higher layers the complexity of the controllers and optimisers increase often with a much lower sample rate.

Most control systems include some type of supervision. The supervision is present at each layer in the control hierarchy for validation of sensor and control signals. Furthermore the supervision must be able to monitor critical parts of plants for safety issues. The supervisory system must be able to react fast when potentially hazardous situations are about to occur. Hence, the tests that the supervisory system performs must be simple, accurate and easy to carry out.

[1] Now this division is a separate company called NovoZymes

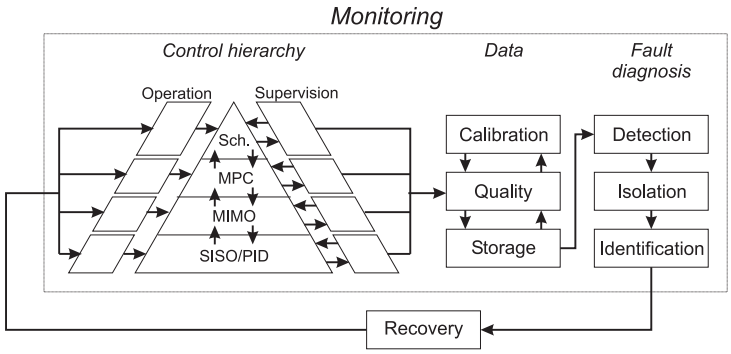


Figure 1.1. Monitoring of a chemical plant consists of multiple elements. The control hierarchy is the classical control system with its SISO and MIMO controllers together with modern MPC controllers and scheduling. Data are routinely obtained from the process, which must be processed and stored efficiently and accurately. Fault diagnosis consists of detection, isolation and identification of faults. For most chemical plants automatic recovery of faults is not possible or is very difficult to implement. Hence, manual intervention is necessary to recover from a fault.

1.3.1 Data Quality

Signals and data are important in the control hierarchy as they provide the control system with inputs that are subsequently used by the controllers to calculate control signals. The data are often stored in databases for later analysis. The quality of the stored data is very important as it provides the basis for the later analysis of the plant behaviour. The actual analysis that will be performed on the data may not be known at the time when the data are stored. This must be taken into account when the specification for the data collection and storage is constructed.

Data must be sufficienly accurate, hence suitable calibration procedures must be developed that meet these requirements.

Data must be available for later retrieval. It is important that the data is readily accessible and can be retrieved fast when the analysis of the data is to be performed. A suitable sample time must be chosen based on criteria for the subsequent analysis. Lowering the sample time will increase the amount of information obtained about the process, but will at the same time increase the storage requirements as well as transfer and processing time. Data compression may be utilised to remove some of these disadvatages, but some lossy data compression schemes may introduce artifacts in the data [Watson *et al.*, 1998]. Thus, the storage and handling of data is nontrivial when one is considering that the purpose of the data gathering may be unknown at the time where it is stored and that the data must be sufficiently descriptive even in the event of a fault in the system.

1.3.2 Fault Diagnosis

Fault diagnosis consists of three steps: Fault detection, isolation and identification (FDII). In order to detect a fault it is necessary that the fault affects, directly or indirectly, the measurements. Using a model, which is called the normal model, it is possible to compare the behaviour and state of a process with the expected normal behaviour of the process. When the process in running *in control* small deviations from normal are expected. If the process is *out of control* the deviations are large. Therefore, if statistically significant deviations from the normal model are seen a fault has been detected. The next natural step is to isolate the fault if possible. When a large number of measurements has been obtanied from a large and/or complex process often only a few of these measurements deviate from the normal when a single fault is present. Hence, these variables can be isolated in most cases. Often this isolation says very little about the physical cause of the fault and some interpretation is necessary. This last step in the analysis of the fault is termed identification. For processes where a detailed model is available it is possible to automate the identification step. This also requires that the model covers the system when it is in a faulty state. For many processes where such a detailed model is not available human interpretation of the fault is needed and hence the identificaton and fault recovery steps are performed manually by the process operators.

1.3.3 Fault recovery

When a fault has been detected and the effect of the fault is damaging to the process equipment or product quality the fault must be eliminated. This is fault recovery. When there is severe danger to the process equipment or personnel it may be necessary to shut down (parts of) the plant. This is of course undesired. Therefore it is important to detect faults early such that faults may be eliminated before they can pose a threat to the process.

In many cases a fault can be eliminated by adjusting process parameters such that the process may continue to operate. This may possibly be at a lower production capacity than for normal operation, but such a condition is often perferred over the situation where product quality is sacrificed.

1.4 Modelling batch processes

The batch process poses some interesting problems from a theoretical point of view. Batch processes are systems that run within a finite time span (the duration is usually pre-specified). The process is often carried out repeatedly with the same or similar recipes. These constraints can be exploited in the structure of the model. Batch processes are generally modelled as nonlinear and/or time-varying processes using nonlinear models. This time-varying behaviour lead to mathematical and numerical complications when controllers and optimisation

schemes are developed.

The data used in this project will be process data from production plants. Process data is defined as data that are collected from processes that are running according to a standard recipe. A recipe is a scheduled specification of how to operate the process (raw materials, initial conditions, when to switch operating modes etc.).

Batch processes are inherently time-varying nonlinear processes and model types do exist that are based on these qualities. Unfortunately, when the model type gets more complicated, possibly offering a more detailed and accurate description of the process, the demands on the data quality and the excitation of the process inputs while the data are obtained are significantly increased. Since such demands are not expected to be met in this project only variations of linear models will be considered.

Industry needs information in order to make the right decision in time. Decisions need to be made on many levels. In this thesis the focus will be on decisions that are directly related to the production process.

Examples of such decisions are

- How can bottlenecks be avoided?
- What is the optimal recipe for a batch process? What microorganism and substrate should be selected and what are the optimal operating conditions (pH, temperature, feeding strategy)?
- When should a process be (re)optimised or redesigned?
- How should disturbances be rejected during operation of the process?

These questions can only be answered correctly if sufficient information is available. Some of the questions require detailed knowledge of the entire production line and the processes involved. Redesign of processes and process equipment is not something that can be entrusted to an automatic system whereas disturbance rejection is something that routinely is automated due the immediate effect of such disturbances.

The goal for industrial processes is to *maximise profit* by developing stable and reproducible processes. It is a goal to *decrease variability* of the processes and ultimately reduce variability of the product. This will allow scheduled processes to run on time so the product can be delivered within specifications to the customer on time. Another way to say this is to say that the processes must be *robust* towards disturbances.

Information is needed in order to make decisions. The information is obtained from the process by introducing measurements where relevant for operation of the process and in order to make supervision and control of the process possible. Measurements from the process alone will not suffice. It is also necessary to know how disturbances (either intentional or unintentional) will affect the future state of the process. To fulfil this need a model of the process is required. For many industrial processes such a model exists only in verbal

form in the minds of the process designers or process operators, which in many cases suffice for successful operation, but is a poor basis for automation and optimisation of the process.

In order to automate process operation design models must be available in a form that can be handled by the process control system. Such models may either be based on first principles (mass and energy balances) and detailed knowledge of the process and its equipment or the models may be developed utilising process data for modelling static or dynamic relationships in the data. These two fundamentally different approaches are illustrated in figure 1.2. Any model provides an approximate representation of the process behaviour. For first principles models the approximations are mainly given through the underlying assumptions behind the model formulations. For data driven models the approximations are given through the data quality and the model structure. Different models for the same system can co-exist depending on the purpose of the model. Requirements for accuracy, calculation time, structure of the model, dynamic and static capabilities and available data determine which model(s) to develop and use for a given application. When different models are developed it is essential to ensure that they are consistent.

First principles models of biochemical systems have been built for many systems and microorganisms, but they are difficult to implement in large industrial-scale plants due to problems with accurately describing large scale phenomena, complicated equipment and the dynamic behaviour of the process in general and the microorganisms in particular. Such model often take the form of ordinary differential equations

$$\begin{aligned} \dot{\mathcal{X}}(t) &= f(\mathcal{X}(t), \mathcal{U}(t)) \\ \mathcal{Y}(t) &= h(\mathcal{X}(t)). \end{aligned}$$

It can be time consuming to obtain the functions $f(\cdot)$ and $h(\cdot)$, but once available they may constitute a valuable tool for the process engineer for the pur-

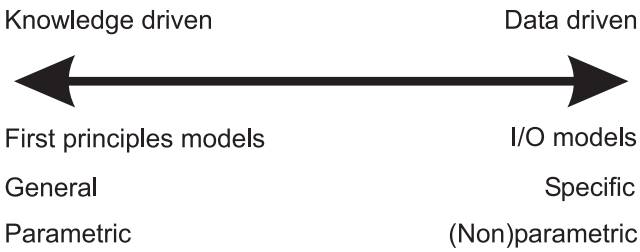


Figure 1.2. Knowledge driven and data driven models represent opposite views on modelling systems. Although the knowledge driven approach and the data driven approach takes their basic assumptions either from prior knowledge or from process data, the final model(s) may be very similar in the static and dynamic behaviour they attempt to describe.

poses of model analysis, simulation, control and optimisation. Often these functions contain unknown parameters which must first be identified using experimental data.

Measurements from industrial processes may be unsuitable to directly support the use of first principles models. States can be impossible to measure and the sample time may be too long and the noise level too high to enable reliable parameter and state estimation. The nonlinear programming problems arising from parameter estimation, model analysis and process optimisation can constitute a significant challenge in developing and using these models.

Despite the associated problems there is great potential in using process knowledge and first principles models due to their predictive power. However, the task of developing first principles models becomes most demanding if one adds that industrial companies often have many diverse processes (types of microorganisms, recipes and process equipment) and changes may be regularly introduced into the process that would need to be modelled.

Models can be built based on data. This approach is called *data driven modelling* whereas first principles modelling may be termed *knowledge driven modelling*. When developing data driven models process knowledge enters when one has to consider which measurements to use and what data should be used in data analysis. The process data stored in industrial data bases are believed to be a virtual gold mine of knowledge as it describes how processes are actually being run and which problems and features that the process possesses. If the patterns in the data can be deciphered and essential information can be extracted a model may be built that can describe the behaviour of the process. Such a model may be used in applications in place of first principles models for e.g. process control, optimisation and perhaps allow for process design. However, one must note that models built on data are generally only applicable within the same operating region as covered by the data used for model development.

Models can either be linear or nonlinear and may have a low number of (physically interpretable) parameters or have many (1000s of) parameters that have no direct physical meaning. General linear models used in multivariate statistics have a low number of parameters that can be used to model simple linear relationships. Artificial neural networks are nonlinear models with a complicated structure that may have a large number of parameters. These parameters will generally not be physically interpretable and the model is termed a black box model. Models that have physically interpretable parameters are called *parametric models*. When the parameters have no clear physical meaning the model is called *non-parametric*. In this case often the number of parameters is large. The methods investigated in this thesis are non-parametric.

In this thesis statistical models based on principal components analysis (PCA) and partial least squares (PLS) will be described. These models are linear, but contain a large number of parameters. Input/output time series models for batch processes will be developed. These models utilise the causal relationship

that exists between measurements at different times during the batch.

The result of the data analysis whether it is a summary statistic or a complicated model must be put to use in order to be valuable. The application of the analysis result is to support decision making. E.g. an immediate decision in the case of a fault diagnosis or process control application.

1.5 Literature review

This thesis focuses on two areas of research. Process chemometrics which is a relatively newly developed data analysis method and dynamic process modelling which has a long history of theoretical development and practical use

The history of of chemometrics dates back to an article by Wold, which often is cited as the first chemometrics article [Wold, 1966]. This article has been followed up many times by his son [Wold, 1995; Wold *et al.*, 2001].

A general introduction to chemometrics is provided in [Martens and Næs, 1989]. An early tutorial can be found in [Geladi and Kowalski, 1986]. More specialised and theoretical results are given in [Höskuldsson, 1994, 1995, 1996]. A book devoted almost entirely to process chemometrics is the book by [Chiang *et al.*, 2001]. This books covers the use of PCA, PLS and CVA as well as some knowledge based methods, which are not considered in this thesis.

The use of process chemometrics methods for fault diagnosis and process monitoring dates back to [Kresta *et al.*, 1991]. These method are multivariate in contrast to early methods for statistical process control that were univariate. Recent reviews are given in [Wise and Gallagher, 1996; Çinar and Undey, 1999; Louwerse and Smilde, 2000; Kourti, 2002]. A review of some early methods for univariate statistical process control of batch processes is given in [Al-Salti and Statham, 1994]. Multivariate metods for batch processes are treated by these articles [MacGregor and Nomikos, 1992; Nomikos and MacGregor, 1995; Martin *et al.*, 1996; Bakshi *et al.*, 1994]. Recent examples of applications of process chemometrics for fault diagnosis can be found in [Tates *et al.*, 1999; Neogi and Schlags, 1997; Gregersen and Jørgensen, 1999].

There has been some development that has tried to extend the linear methods most often applied in chemometrics with nonlinear terms or the application of entirely nonlinear models e.g. artificial neural networks. Nonlinear regression methods related to chemometrics can be found in [Wold *et al.*, 1989; Baffi *et al.*, 1999b,a]. Artificial Neural Networks (ANN) are often applied as a nonlinear regression method on chemometric data. A general introduction to neural networks can be found in [Haykin, 1994] and a special article for chemometricians can be found in [Svozi *et al.*, 1997]. A review of ANN's used for fault diagnosis is given in [Zorriassantine and Tannock, 1998].

A key subject of this thesis is to include dynamic information in the data driven models of batch processes.

An introduction to state space models can be found in [Rugh, 1996] that em-

phasises on classic control theory, stability analysis etc. A review of continuous-time identification is given in [Unbehauen and Rao, 1998]. Ljung presents the classical introduction to discrete linear time-invariant and time-varying input/output modelling in [Ljung, 1987].

Modelling of batch processes using input/output time series models is not covered in the literature in great detail. However, some information can be found in [Russell *et al.*, 1998]. Masses of literature on I/O modelling of time-varying systems do exist [Dewilde and van der Veen, 1998], but little space is devoted to batch processes and is therefore not directly applicable to this process type.

The use of subspace identification methods on time-varying systems is covered in [Verhaegen and Yu, 1995; Liu, 1997]. The correspondence between state space models for continuous systems and chemometric tools was investigated in Wise [1991]. The use of Partial Least Squares (PLS) and Canonical Variate Analysis (CVA) for modelling continuous processes is treated in [Simoglou *et al.*, 1999; Negiz and Çinar, 1998; Çinar and Undey, 1999].

Reviews of control of batch processes in general in given in [Berber, 1996]. The control of fermentors is reviewed in [Rani and Rao, 1999; Shimizu, 1993; Jørgensen and Jensen, 1989]. These papers do not base their work on data driven models.

1.6 Scope and Outline of the Thesis

1.6.1 Hypothesis

The goal of the thesis is to investigate how existing industrial process data can be utilised in a systematic way on a regular basis to improve batch process operation and understanding.

1.6.2 Achievements

The thesis is divided into three parts. The first part discusses the use of data driven models and proces chemometic methods for proces monitoring and emphasises the need for monitoring the data quality of the collected data. The matrix notation used in this part of the thesis facilitates the comparison between methods which is carried out in the last part of the thesis. The second part describes a new method for dynamic batch time series model identification using input-output data and its direct applicability for monitoring, process control and optimisation. The developed modelling methodology uses the special structure of dynamic batch data and regularisation methods when estimating the parameters. The last part of the thesis compares these two different modelling techniques and presents an improved method for process monitoring that takes the process dynamics explicitly into account.

The body of this thesis, contained in chapter 2 to chapter 7 is a collection of articles. A drawback of this is that some descriptions return in every chapter and are therefore redundant. On the other hand, every chapter is self contained and can be treated as such: it is not necessary to read all chapters preceeding the chapter one is interested in as most of the information is repeated making it easier to read each article independently.

Currently only the material in chapter 4 has been published [Gregersen and Jørgensen, 1999]. Chapters 5, 6 and 7 are planned for publication.

1.6.3 Scope of the thesis

The investigations require that methods are assembled for finding interesting patterns in data and extracting the information that is needed to support the process engineer or process operator. It has not been the goal of the project to build tools (software) that are suitable for production use, but rather to look into prototype methodologies for data analysis and modelling.

Only a very limited description of the biological processes is given in this thesis. The methodologies used in this thesis will of course be influenced by the application, but the results are not limited to fermentation processes alone. They are applicable to batch processes in general.

Since only normal operating plant data is used in this thesis it will only be possible to describe process behaviour that is close to the normal operating region. No special experiments have been carried out in this work in order to obtain data from special operating regions or to excite the process. The focus will therefore be on *fault diagnosis* and *dynamic modelling* assuming that the modelled process can be operated in a narrow operation region.

The focus of the thesis is on modelling fermentation batch processes using linear models. Whenever possible the stated problems and solution methods will be formulated using matrix notation. Other alternatives are to write expressions using operators or to use a vector notation and summations. The matrix notation is used to keep expressions simple and compact. Notation and basic linear algebra are described in appendix A. Statistical background information is given in appendix B.

The numerical computations performed for solving the problems in this thesis are performed using MATLAB [MATLAB, 1999]. The programming routines themselves are not reproduced in this thesis.

1.6.4 Outline

Chapter 2 discusses in greater detail the data flow illustrated in figure 1.1. In particular how to obtain process data and how to perform data validation. Data analysis may be divided into two parts: The first part consists of data visualisation and calculation of summary statistics. The second part consists of data selection and modelling. The developed models may be applied within

the fields of process monitoring and process control. The chapter ends with a description of some tools developed for data mining and a summary.

Chapter 3 describes methods for data mining and fault diagnosis. These methods are based on chemometrics methods, which also are particularly useful for analysing spectroscopical data for analytical chemistry. Since the data analysed using chemometric methods in this thesis are process data the methodology is termed *process chemometrics*.

Chapter 4 is an article describing the use of the process chemometrics methods on a real data set from a set of fed-batch fermentations from an industrial plant.

Process chemometric models are static models although they can be given a dynamic interpretation. Dynamic input/output models can be built instead. The model structure used is based on linear time invariant (LTI) modelling. By using this relatively simple structure it is possible to utilise the vast amount of theoretical results available for LTI models. Batch processes require a special model structure compared to the conventional model structure used for continuous processes. Chapter 5 introduces such a model structure. The focus in this chapter is on identification of the model.

The methods used for process chemometrics have much in common with batch LTI methods. By combining both principles models can be built that utilise the structure of the data, the causality of the process and the input/output behaviour. These relations are outlined in chapter 6 and a scheme is given for how to integrate the correlation approach of modelling with the causal LTI modelling for fault diagnosis.

This chapter shows that it is possible to develop dynamic batch models that can be used for process monitoring and that the previously described models (in chapter 3 and 4) do not make use available knowledge about process dynamics even though the data used for developing both types of models is the same.

Chapter 7 shows how the identified models can be used for process optimisation and model predictive control by directly using the model formulation resulting in simple batch operation schemes. This chapter uses simulated data in its examples.

The thesis ends with conclusions and a discussion of future work in chapter 8.

2

Data Driven Modelling

Data mining for data driven modelling is a versatile combination of tools for developing static and dynamic models of complicated systems and processes. When modelling is based mainly on data it is important to focus on the data quality and the suitability of using existing data for modelling. This chapter describes in detail the steps involved in obtaining data and models in the iterative procedure of exploratory data analysis also known as data driven modelling.

This chapter describes practical concerns associated with collecting and analysing ordinary process data that are obtained from a normally operating plant where especially the operation, but perhaps also the data collection are not specifically designed to be used for modelling. Process data may have severe faults that must be dealt with before submission of the data to any modelling algorithm. Some help in dectecting and removal of faulty data can be gained from computational tools, but hard work is required to select data which are suitable for the modelling task and this selection and screening procedure is often labour intensive and several iterations of data selection, modelling and validation must be performed.

There is a large amount of literature available that treats the use of large databases for data mining. However, few books and articles discuss process data, but are entirely devoted to business data. The distinction between these two types of data is treated in [Fayyad *et al.*, 1996a].

Numerous methods exist for exploratory data analysis. Classification is discussed in [Wang and McGreavy, 1998]. Pattern recognition and dynamic trend interpretation is treated in [Saner and Stephanopoulos, 1992; Bakshi and Stephanopoulos, 1994a,b]. Rule based system may be developed where the extraction of the rules may be data based [Ignova *et al.*, 1996; Mulholland *et al.*, 1995]. For numerical data a substantial number of statistical methods are developed [Glymour *et al.*, 1997; Little and Rubin, 1987]. For multivariate data mining problems in chemical processes the methods of process chemometrics may be used for process modelling and fault diagnosis [MacGregor and Kourti, 1995]. Prior to data analysis is the step of data preparation and data cleaning [Pyle, 1999]. The textbook by [Chiang *et al.*, 2001] deals with the area of process chemometrics as well as analytical and knowledge-based methods with

examples from chemical engineering.

This chapter describes the various phases of the iterative modelling procedure of building a statistical process model based on process data.

The purpose of this chapter is to clarify key steps in the iterative data driven modelling cycle with emphasis on the importance of maintaining data quality on a level that is suitable for the future application of the data to be succesful. This chapter contributes with the important message that calibration of measurements and data preparation are essential steps that must be given special attention throughout the collection and analysis of the data. These issues are often neglected by the literature that simply assumes that the data quality is sufficient for the task at hand and/or any data pre-processing has been carried out prior to the "real" analysis.

The process of data analysis is an iterative process where many tools are necessary in order to extract information from the gathered process data. The methods used will depend on the purpose and goals of the analysis, but since the data analysis is an exploratory discipline the goals may change during the analysis phase. Such changes of goals may in turn result in extra conditions and requirements on the data quality.

This chapter defines the data mining cycle in section 2.1. This cycle is a iterative procedure for data mining involving the many steps one has to go through in order to perform data driven modelling. The many steps of the cycle are explained in the subsequent sections. Selection of the measurement techniques is important to obtain the optimal data for the data mining. This issue is described in section 2.2. Data must be properly validated before they can be used for data processing. Data validation is described in section 2.3. The data analysis step which contains visualisation, summary statistics and data mining is dealt with in section 2.4. Various applications such as monitoring and process control are described in section 2.5. A discussion of this chapter is presented in section 2.6. Conclusions are given in section 2.7.

2.1 An introduction to process data

Process data contains measurements of physical quantities such as mass, volume, temperature etc. Two types of process data exist that have different features:

Data from (designed) experiments. Small scale or large scale, special operation of equipment (excitation of inputs or change of operating range within the possible window), additional expensive or difficult to obtain measurements, shorter sampling time than for normal operation.

Process data. Normal operation, fixed sample time, database created for future reference.

The difference between these two types of data is that data arising from experiments are developed for a specific purpose and the scientist performing the experiment therefore has a precise idea of what analyses to perform on the data. Process data are collected at various points in the process, but the data are traditionally used for (single loop) controllers and simple diagnostics purposes and normally not combined for knowledge extraction.

Data mining is a rapidly growing area concerned with data analysis methods used to explore more or less structured databases in order to find useful information. The discipline is also called knowledge discovery in databases [Fayyad *et al.*, 1996b,a].

The purpose of data mining is to extract information from large databases and summarise the content to provide the user with high level knowledge. This knowledge may be used to change the future process data either by affecting the measurements or by changing the operation of the process by using e.g. experimental design or process control. Process data can experience missing data, high noise level and bias. Often multiple sample rates are used, which makes the use of the data difficult. One can in general not conclude that process data are directly suitable for modelling since measurements that are necessary to support modelling may not be implemented. Data are often just collected to give an overall impression of the operation of the plant, but little attention is given to support e.g. on-line monitoring of mass or component balances.

Thus, it can be a complicated task to perform data analysis and data mining on industrial processes.

The data mining cycle is illustrated in figure 2.1. This cycle schematically illustrates the iterative process of obtaining measurements, data preparation, modelling and various applications. At all stages of the cycle prior knowledge, assumtions and goals may affect the analysis. If the goals cannot be fulfilled with the current quality, type or quantity of measurements it is necessary to improve these factors before performing the analysis and modelling again. One may also come to the conclusion that improvements of the measurements or models are not feasible due to e.g. technical or economical reasons and then the goals have to be reduced to a realistic level.

The following sections will explain the four parts of the modelling cycle in more detail.

2.2 Measurements

In order to make decisions, whether they are directly related to the operation of a process or it is a management/scheduling decision, it is important to have the best information for the task.

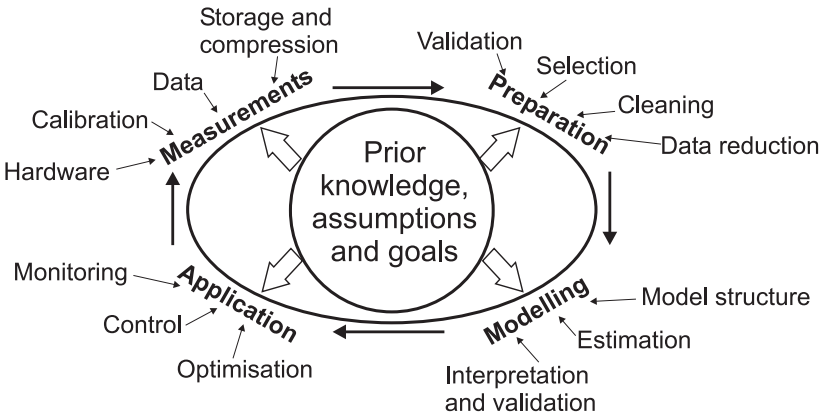


Figure 2.1. Data mining cycle. Measurements, Preparation, Modelling and Application depend on each other and must be compared to the overall goal. Prior knowledge and assumptions will initially define the goal, but the goal may change as the analysis of the data takes place.

2.2.1 Measurement devices and soft sensors.

This means that data must be collected at various points in the process such that it can provide the basis for the information needed for the decision making. When the data are collected over long time massive amounts of raw data are collected. This amount of data will usually be too large to be handled as it is and the structure of the data may be too complicated to directly support the decision making. Data mining and finding summary statistics are necessary in order to extract the information from the data in a format which is suitable for further processing. Data compression and projection methods are often used to reduce the amount of work that has to be carried out on the data and provide simplified data to the modeller. By combining information contained in data with prior knowledge a higher level of understanding of the underlying process behaviour can be obtained. Thus, providing an improved basis for decision making, e.g., concerning control actions.

One important task is to determine what kind of information is needed to support specific decisions. In order to make a process line run efficiently and profitably a limited number of on-line measurements will be implemented and only few off-line measurements should be made since they may require substantial amounts of man-power and expensive analysis equipment. Off-line measurements may also have a significant delay that can pose severe limitations to their use.

Planning and design of a production line require a number of decisions concerning:

- Which physical extensive and intensive variables to measure?

- What measurement technique to use (accuracy (bias), noise level (variance), cost, reliability, sample time, maintenance)?
- Relevance for the application?

The selection of what to measure requires careful selection. Some measurements are most suitable for on-line control systems. Some measurements are necessary for detecting and handling faults and some measurements are suitable for evaluating the overall performance of the process. Often quality variables are not directly measurable in their proper context. E.g. in the case of making enzymes for detergents the quality of the enzymes is not evaluated in a washing machine, but assays are made instead that mimic the result of having used a washing machine.

Reliability is an important issue. One should choose simple reliable measurement techniques instead of advanced measurement devices that easily may fail. This selection criterion will facilitate the use of the measurements in e.g. control systems and for supervision. In the case of a fermentor one has one type of off-gas analyser to measure the consumption of oxygen and the generation of carbon dioxide by the microorganisms. This type requires the handling of the off-gas stream and an expensive mass spectrometer to perform the analysis. The sampling time may be low because the mass spectrometer is shared between several tanks. One could obtain essentially the same type of information (for a specific process) from measurement of the heat generation, which can be measured by relatively simple and reliable temperature and flow measurements of the cooling water inlet and outlet streams [Duboc, 1997].

More advanced estimation methods are sometimes advantageous. Secondary variables such as pH and temperature are relatively easy to control, but has a weak connection to the variables of real interest—concentrations and yield. For many biotechnological applications quality variables and performance characteristics are the variables of real interest, but these are often not directly measurable and are therefore unsuitable for a direct implementation in a control system. One way of handling this problem is either to invest in complicated analysis equipment or to develop soft sensors.

Soft sensors can be implemented in some cases in order to utilise the data already available in the plant on-line as an addition to or substitution for expensive off-line analyses. A soft sensor is an algorithm that can infer unmeasurable variables from the available measurements. This can be done by combining different measurements with a soft model (data fusion) or by combining data and process knowledge (state estimation). In both cases some sort of a model is required of the system that describes the relationship between measured and unmeasured variables. Both static and dynamic process models can be used for this purpose. Soft sensors require accurate and reliable models or calibrations to work and one must balance the cost of maintaining models and calibrations against the cost of adding measurement devices to the plant.

It is an important task to be able to estimate key variables for which measurements are not available at all or are only available with a very long sample

time. Using the available measurements and a model the current state of the process may be estimated (see figure 2.2). If desired, the model may also be used for the prediction of future states.

Data for the control system for a chemical process are often sampled with a sampling time in the order of seconds. Even with a small number of measurement devices a massive amount of data will be collected when data are sampled every minute or hour. For monitoring chemical processes it is usually sufficient to use a sample time in the order of minutes. When data are stored in a database they are often resampled to a sample time in the order of many minutes or hours. Filtering of the data is thus necessary to avoid aliasing if the data later are used for dynamic modelling [Ljung, 1987].

2.2.2 Tools for data storage, retrieval and analysis.

Data do not only have to be collected it must also be stored and analysed. This requires computational tools in three areas

- Data storage and retrieval.
- Data analysis and visualisation software.
- On-line software for process monitoring and process control.

It is not only important that measurements are identified and the sensors and transmitters are set up. The process control system must be able to handle the massive amounts of data and be able to store the data in an accesible database.

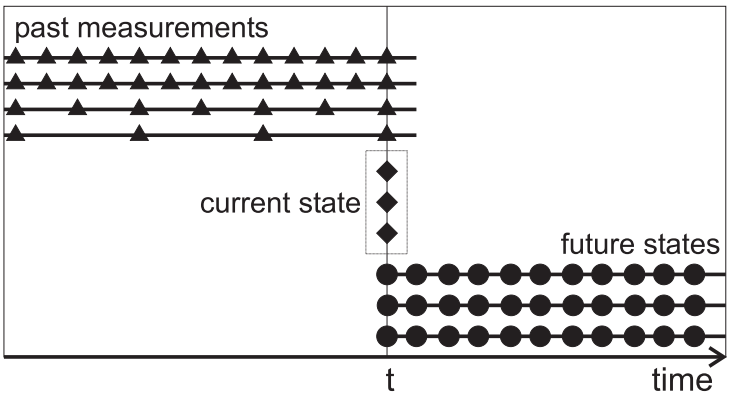


Figure 2.2. Estimation and prediction from measurements in batch control. Measurements may be sampled with different sample times and delays up until the current time. The current state value is estimated using the past measurement and a model of the system. If the model is suitable for prediction purposes it may be used to predict the future state values of the system. As the batch progresses the estimation problem becomes larger because there are more past measurements to consider whereas the prediction horizon decreases.

The raw process data are usually filtered at several points in order to remove noise. Often data are not stored in the database as often as data are sampled by the process control system and extra filters must be applied in order to avoid aliasing.

Data compression techniques may be employed to save storage space. The data compression can either by *lossless* or *lossy*. The lossless storage method stores data in a way such that they may be retrieved in its original form. This is the storage method used in traditional database systems. Storage space can be saved if the compression method is allowed to be lossy. Lossy compression techniques only store changes in the data that are larger that some threshold. Such thresholds can be specified in the time domain or frequency domain.

For the time domain it is common to use either a variance measure or some hard bounds to check when the process variable is almost constant or has almost constant slope. When this is the case measurements may be discarded and data points are only saved when the process variable is experiencing large changes. Since many process variables either do not chance often or do not change much a large amount of storage may be saved using lossy compression.

When filters and compression techniques are applied to the data it is imperative that important features of the data are not lost. Since compression techniques usually are applied as univariate filters operating on one variable at a time the mean or covariance matrix may be seriously affected by lossy compression [Watson *et al.*, 1998].

If the thresholds for deciding when a variable has changed significantly are not based on the covariance matrix some variables may be overly compressed and information from these variables will be removed whereas other variables may be virtually unaffected by the compression. The overall effect of the compression process will then be the potential corruption of the mean and covariance of the collected data, which may in turn severely affect the subsequent modelling steps. Furthermore, lossy compression techniques are designed and tuned to work under normal operating conditions. When a fault occurs in the plant the parameters for the compression algorithms may not be optimal for storing information about the fault. E.g. slow drift of the system may not be detected early due to the removal of such information.

Data compression techniques can be used as one of the ways to facilitate faster sampling of (some of) the process variables. If the sampling interval is decreased for the compressed signal compared to the uncompressed, slowly sampled signal the advantages of obtaining more high frequency information may to some extent outweigh the disadvantages of introducing artefacts by the compression.

The design problem of these compression algorithms must involve the multivariate properties of the process data and the need to document normal as well as abnormal process operation. Since stored data will serve many purposes in future analyses not envisioned when the data are collected and compressed such a compression design issue is not a straightforward task.

When data are collected it is important to evaluate the data. Data validation is the subject of the next section. The process may experience upsets that will affect the model in an undesirable way or the data may contain outliers. Systems must be set up that can handle these situations so the model development can be performed on a suitable representable set of data for the intended process behaviour.

2.3 Data preparation

The data collection itself is often only a little part of the data handling. The major part of the initial data analysis will be on data validation, selection and data cleaning. These steps lead up to the application and may be a part of the modelling step, but are initially separate steps where suitable data is selected and processed for the purpose and application as shown in figure 2.1. This step must be an iterative process where data and process knowledge are processed together to form proper models of the system that support the purpose of the application.

Data are sampled from the process, usually at regular sample intervals. It is important that the collected data represent the relevant dynamics of the process and that the measurement do not have gross measurements errors.

On-line process data such as pressure, temperature, pH, weight, flow etc. are obtained regularly from unit operations in the production line. These measurements are the basic sources of information about the operation of the process. These measurements are used in control loops to control the process at the desired setpoints or trajectories and to drive the process toward optimal performance and to compensate for disturbances that enter the process. Quality variables are obtained that are directly related to key components that describe the state of the process. For a fermentor an 8 hour or 12 hour sample time may be chosen for these quality variables when the batch duration is more than 48 hours. Quality measurements often require time consuming laboratory analysis which introduces a delay. These measurements are therefore termed off-line measurements. These off-line measurements are used for the estimation of the performance of the process. Such measurements are expensive, but important in order to ascertain if the process is in control and whether the expected performance of the process can be obtained. Quality data are finally combined into data to support management. This can be data that demonstrates the efficiency and profitability of the process or suggests when optimisation of the process design or operation is necessary.

Data storage and retrieval methods must be designed for fast retrieval and processing such that accurate and timely data analyses can be performed. Although there is not a clear boundary between management data and process data, business data usually have longer sample time than process data as illustrated in figure 2.3. Raw process data will usually be used for the control of the process with sample times in the order of seconds or minutes (e.g. the

control of a valve). The setpoints of this control layer are determined from the layers above based on the quality data, prior knowledge and the selected control structure.

It is important that the quality of the data is evaluated and suitable specifications are defined based on the intended application of the data. Depending on the purpose of the data very different requirements can be formulated for e.g. sample time, noise level, reliability etc. Furthermore, calibration routines and maintenance programmes must be set up to assure that the measurement devices and transmitters are kept in good quality condition. These requirements must constantly be monitored in order to assess if they continue to comply with the application. Only by observing data quality in context with its use a successful application can be made [Orr, 1998].

An example of such a data quality problem may be seen in figure 2.4. Here dissolved oxygen tension measurements (DO) are shown. The measurements are obtained from a bacterial fermentation using *the same recipe*. It is obvious that the curves are very different. Small differences are expected between batches. However, the differences shown in figure 2.4 are caused by failing calibration procedures rather than a real difference in the operation. For data mining purposes there is not much information in the data seen in figure 2.4. Better calibration procedures were later introduced to obtain more accurate quantitative information, which increased the repeatability immensely.

Outliers and missing data may also cause problems. There is not a unique definition of what an outlier is in the literature. An outlier is here defined as one or more entries in a database that has been stored with a value that is obviously outside its normal range or otherwise unrealistic due to violation of conservation laws or physical constraints. The criterion can (and should) be evaluated in a multivariate sense also taking dynamics into account where possible such that known correlations and dynamics are exploited in the analysis of the data.

There are two ways of handling outliers and missing data [Little and Rubin, 1987]:

Detection and imputation. Initially outliers are detected. Then the data

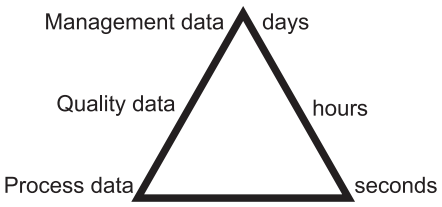


Figure 2.3. The sample time for collected data and control loops vary much depending on the use of the data and on the extent that the data processing can be automated.

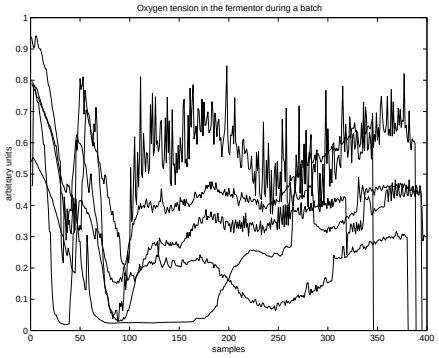


Figure 2.4. Dissolved oxygen tension measurement (DO) in a fermentor for several batches using the same recipe.

values are imputed with a better representation of the data based on the other measured variables and a (rough) model of the normal behaviour of the system [Little and Rubin, 1987].

Robust methods. Methodologies exist that automatically suppress the effect of outliers and base the estimate mainly on good data. Unfortunately such methods are very computer intensive and are not widely used [Rousseeuw and Leroy, 1987].

It is important to identify data that has been misrecorded such that subsequent modelling steps do not attempt to model irrelevant features in the data. It can also be convenient to eliminate parts of the data that are irrelevant to the current analysis e.g. data obtained during process startup/shut down or when the process is experiencing faults that should not be modelled. The relevance of the data will of course depend on the application of the analysis and therefore there will be varying definitions of what an outlier is, depending on the application of the data and not only on the data itself.

In many cases it is in the pre-treatment phase that most of the time and effort is spent. When the data have been thoroughly analysed and cleaned for abnormalities it is a fairly quick exercise to perform the modelling [Pyle, 1999].

In all cases it is important that data is not just eliminated because they *appear* to be outliers. There is often a cause for seemingly abnormal data. If the fault is in the process itself and not only in the data collection the fault may possibly have an impact on the product and the problem must be identified and dealt with. As it is often not possible to reconstruct an outlier or otherwise missing item in the database these problems must be dealt with in a timely manner.

Smart sensors may be a way to automatically overcome some of the problems with validation of the data. Smart sensors are to some extent able to detect degradation of the sensor or when recalibration is necessary [Leahy *et al.*, 1997].

Furthermore, smart sensors should provide an uncertainty estimate. These type of sensors constitute an emerging technology and is now becoming more widely available [Clarke and Ghaoud, 2002].

2.4 Modelling

The modelling step is the step where the massive amounts of process information available in the collected process data is condensed into models that are used to describe the process behaviour and these models may subsequently used for fault diagnosis, process simulation and process control and optimisation.

Data driven modelling is a method to obtain new knowledge, i.e. insight, through the use of data. The explorative nature of the discipline means that data has to be processed in an iterative manner until the right combination of data and model type has been found. All data handling and data mining applications benefit from various graphical visualisations. 2-D and 3-D plots of the raw data and summarising statistics can provide the analyst with the initial analysis and help with pointing out interesting parts of the data and help identify outliers in the data [Tufte, 2001].

Statistical software exists that can summarise the data into e.g. mean and variance. These summarising statistics help the identification of normal and abnormal operating regions and can furthermore be used for a rough identification of outliers.

2.4.1 Data analysis

Data mining requires massive amounts of data to extract information. If the data extraction is successful new knowledge and understanding can be formed as illustrated in figure 2.5. Such knowledge may be condensed and formulated using equations or rules. In such cases we say that we have a mathematical or statistical model of the system.

Using basic data analysis tools and prior process knowledge it is possible to extract the interesting and useful parts of the data and thereby form a representative *normal data set*. This normal data set forms the basis for subsequent data analysis. Several models may be built during the data analysis and modelling step and often only a few models are selected for a specific application whether it is process monitoring or process control or it serves yet another purpose.

Process disturbances and faults that require a quick response must be handled by an automatic control system. When the problem cannot be handled by a single loop controller or by imposing simple bounds on the variable a model based solution strategy may be advantageous and data mining may provide a means to yield a working model of a complex industrial plant or unit operation.

Prior knowledge may not be sufficient to build a first principles model based on conservation principles and knowledge about process dynamics. Even when

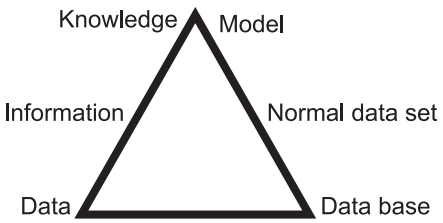


Figure 2.5. From data to knowledge. Massive amount of raw process data may be processed to extract information from the data and eventually form process knowledge. The amount of data is reduced as the modelling is performed.

sufficient prior knowledge is available and a model can be formed a significant amount of work remains to be carried out to tailor the model to the particular system because of experimental design, parameter estimation and model validation that has to be performed.

2.4.2 Modelling tools.

The data and the developed models can be used off-line for e.g. analysis of the performance of the process and for determining whether tuning of the control loops are necessary at the low level of the control hierarchy. At the higher business level models can be used to determine whether process optimisation may be worth while.

The selection of suitable modelling tools for data mining depends very much on the intended application. Statistical software and general signal analysis software can be used to summarise the data. For data mining powerful data handling and data manipulation tools are necessary.

Since data mining is an iterative process with a high degree of interactivity with the modeller the tools must facilitate such interaction. This requires that the computations can be performed fast. This requirement may have been a significant factor in the selection of linear models as being the dominant model type used by chemometricians. Integrated tools for handling the collection and storage of data as well as process documentation, data mining and application of the developed models off-line and on-line are currently not widely available as commercial software. The modelling software most commonly used by chemometrics developers is MATLAB [MATLAB, 1999]. A useful toolbox for performing such analysis is the PLS Toolbox developed by Eigenvector Research. Many commercials tools exist for handling batch recipes, but very few exist specifically for batch modelling. One such tool is SIMCA-Batch On-Line that is made by Umetrics.

Many of the tools that do exist today for data analysis are not tailored to batch process data. Batch processes are special in the way that they are highly nonlinear and nonstationary. The tools must be able to handle that

the conditions at the beginning and at the end of the batch process are very important for evaluating the performance of the batch and that batch processes are repeated many times using the same recipe.

An integrated tool for modelling and data management is necessary in order to efficiently make use of the obtained data and to make the transfer and storage of process knowledge efficient. An integrated tool eliminates the time consuming task of data conversion that is necessary when multiple tools are used and facilitates the reuse of process and modelling information stored in the system. Especially in the case where the modelling and fault diagnosis are to be carried out by non-chemometricians an integrated, user friendly tool is desirable.

2.5 Applications

Numerous applications of process data and developed models do exist in an industrial company. This chapter focuses on two areas. Process monitoring and dynamic models for optimisation and control.

2.5.1 Process Monitoring

Monitoring of processes is a task where process operators assisted by sensors and the process control system supervises the processes and identifies whether the process is operating normally (is in control) or the process has changed from normal operation (the process is out of control). Out of control situations can be caused by sudden disturbances or slow changes to the process equipment e.g. fouling, instability of microorganisms or degradation of a catalyst. The process control system has the task of compensating for these short or long term disturbances. The goal of the automated process design is that the process can be operated even when disturbances affect the system. The presence on the control system also has the consequence that process degradation and faults can persist undetected for a long time before they are observed because the control system can compensate for the fault and hence disguise its presence. By combining the control system with a supervisory system the plant may be operated within normal operating regions for a long period of time with proper warning about when maintenance and fault recovery are demanded [Chiang *et al.*, 2001; MacGregor and Nomikos, 1992; Gregersen and Jørgensen, 1999].

2.5.2 Process Control

Process control systems in manufacturing has the task of maintaining key parameters at a desired value. For continuous operation the values are constant for long periods of time, but for batch operation key parameters vary during the batch. A process control system must be able to handle both types of operation as most modern chemical plants include batch and continuous operation phases. Key values to be controlled are flows, temperature, pH, pressure etc.

These variables are usually controlled using single loop controllers and their setpoints are determined by a recipe for the batch.

The prevailing operating strategy in today's batch processing plants is to operate the batch processes using recipes. This means that batch processes are started with carefully designed initial conditions and are operated using predetermined trajectories for controller setpoints. It is the goal that key variables will then also follow predetermined trajectories that lead to the desired products and byproducts in the desired quantities.

Since the strategy outlined above is sensitive to disturbances both in the initial state and during the batch an improved strategy would be to define control trajectories for key quality variables (similar to the setpoint for a continuously operating system) and have the control system follow these trajectories while imposing constraints on relevant key variables and control signals. This strategy is more complicated as the relationships between states are multivariate. Hence, multi input/multi output control may be required to achieve the desired performance. As the complexity of the relationships between variables increases there is an increasing need for model based control design. For continuous processes the literature on empirical modelling has been developing over decades. For time-varying processes such as batch processes there has been much lower research interest in data based model development.

Automatic control is imperative for the lower layers in the control hierarchy where the sampling intervals are very short as illustrated in figure 2.6. Manual intervention in these areas is not possible within a reasonable time and the control system must therefore be augmented with a supervisory system that perform simple validation of the gathered process data and control signals.

For the middle layers where the sample times are in the order of minutes to many hours it is to be decided by the developer whether normal process control and disturbance rejection is handled by an automatic process control system or by manual intervention. Such decisions must be based on the dynamics of the plant together with the performance and safety criteria of the process. Additionally, the severity of the consequences in the event of a fault must be taken into account.

If automatic control is to be implemented on these layers in the middle then process models are desirable to describe the effect of the control actions. These models must in general be multivariate. However, they are currently not available for many chemical processes dealing with complex reactions in complex media. The lack of multivariate models is a problem especially for biochemical processes where one also has to deal with the complexity of the living organisms involved.

2.6 Discussion

It is important to note that even though data play a central role in the data driven modelling process it is important to utilise the existing process know-

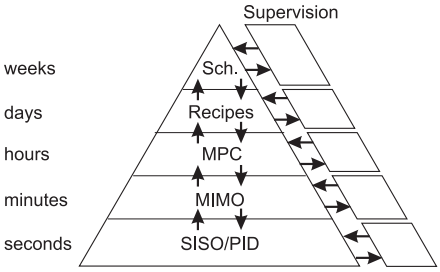


Figure 2.6. Control hierarchy with typical sampling intervals. A supervisory system exists at each level to verify and corrent inout and output data for the controllers.

ledge when collecting and analysing data. This integrated utilisation of knowledge is important both when designing the measurement devices in the plant, but also when data are obtained during experiments or normal operating conditions and later when the developed models are applied to the plant.

The data mining cycle shown in figure 2.1 shows that the modeller must go through many steps and possibly many interations before a model can be turned into a useful application.

After each of the steps in the cycle has been performed it is important to critically evaluate the result based on the assumptions of the analysis and the intended application of the data mining to ascertain that the data quality is suitable for the application and that the underlying assumptions are not violated.

It is a good exercise to include stopping criteria for the iterative process in the assupmtions for the model development. Such a criterion could be based on solving a specific problem or one could choose to stop when the cost of continuing gets prohibitive or one gets out of time. However, often the criteria depend directly on the outcome of the data analysis and hence the goal can change depending on the result of the data analysis.

2.7 Conclusion

It has been illustrated that data mining is a multidisciplinary task, which requires a chain of data processing steps that has to be used iteratively until the relevant information has been extracted from the data. It is important that the quality and reliability of the data is assessed when it is used within the off-line data processing step and even more importantly when the data or the developed model are used for on-line applications.

During data collection and application of developed models the data must be constantly supervised to ascertain that the sensors are working and properly calibrated. During application of the developed models the process operation must be supervised to ensure that the developed models are up to date and that

the data which the developed models are exposed to belongs to the modelled classes of behaviour.

3

Process Chemometrics

Process chemometrics is a newly developed area combining chemometric methods and process data. Process chemometrics is mainly based on multivariate statistical analysis and regression methods that can be used to extract information from process data. The developed statistical models can be used for data analysis (data mining) and fault diagnosis. This chapter gives the theoretical background for detecting faults and isolate the variables that indicate the fault.

Process chemometrics is a field of research that is an off-spring of *chemometrics*. There a many definitions of chemometrics. Taken literately is it a field involving *chemistry* and *measurements*. It will often involve some mutivariate statistical analysis. Other similar areas exist: econometrics, psychometrics etc. All these research areas are collected under the name of *technometrics*.

Different types of data are used for the different types of technometrics, but the methodologies are basically the same. Even for chemometric applications there are wide differences between the types of data that the chemometrical methods are used for. The handling of spectroscopical data involves regression modelling of low noise data that are the result of carefully designed calibration experiments whereas process chemometrics is used for analysing happenstance, noisy data based on measurements of multiple physical quantities, e.g. pH, temperature etc.

Chemometrics can be used for basic data analysis of a data set that is unknown to the chemometrician. Chemometrics can then be used to find interesting patterns in the data that need further analysis. Whether this analysis is performed within the framework of chemometrics or some other modelling type is up to the analyst.

The history of of chemometrics dates back to an article by Wold, which often is cited as the first chemometrics article [Wold, 1966]. This article has been followed up many time by his son Wold. Recently with an overview article about the history and future plans for chemometrics [Wold, 1995]. In this article chemometrics is defined as

> How to get chemically relevant information out of measured chemical data, how to represent and display this information, and how to get such information into data.

This is a rather broad definition, but there is little reason to make it more restrictive.

A few good books treat chemometrics. [Martens and Næs, 1989] provides a general introduction to chemometrics. An early tutorial can be found in [Geladi and Kowalski, 1986]. More specialised and theoretical results are given in [Höskuldsson, 1994, 1995, 1996]. Related to this field are books on multivariate statistical analysis [Jackson, 1991; Mardia *et al.*, 1995].

The use of process chemometrics methods for fault diagnosis and process monitoring dates back to [Kresta *et al.*, 1991]. Recent reviews are given in [Wise and Gallagher, 1996; Çinar and Undey, 1999; Louwerse and Smilde, 2000]. Dynamic modelling, fault diagnosis and expert systems are featured in the textbook [Chiang *et al.*, 2001]. A review of some early methods for univariate statistical process control of batch processes is given in [Al-Salti and Statham, 1994]. Multivariate metods for batch processes are treated by these articles [MacGregor and Nomikos, 1992; Nomikos and MacGregor, 1995; Martin *et al.*, 1996; Bakshi *et al.*, 1994]. Recent examples of applications of process chemometrics for fault diagnosis can be found in [Tates *et al.*, 1999; Neogi and Schlags, 1997; Gregersen and Jørgensen, 1999].

Nonlinear regression methods related to chemometrics can be found in [Wold *et al.*, 1989; Baffi *et al.*, 1999b,a]. Artificial Neural Networks (ANN) are often applied as a nonlinear regression method on chemometric data. A general introduction to neural networks can be found in [Haykin, 1994] and a special article for chemometricians can be found in [Svozi *et al.*, 1997]. A review of ANN's used for fault diagnosis is given in [Zorriassantine and Tannock, 1998].

The purpose of this chapter is to describe the mathematical and statistical background for process chemometrics and how it can be used for data analysis and fault diagnosis. Emphasis has been placed on creating a description that makes use of a compact matrix and vector notation to keep the presentation easy to understand at to later reveal relationships between seemingly different modelling approaches.

The choice has been made to focus on linear methods in order to maintain a simple model structure as a first attempt. These methods are used to investigate if the available process data from one of the production plants of Novo Nordisk would be suitable for process chemometrics analyses and applications.

Section 3.1 introduces principal component analysis (PCA) which is a very general statistical method of analysing data and the underlying process that has generated the data. PCA is a multivariate method that projects data into a lower dimensional space. How to select the dimension of this lower dimensional space is described in section 3.2. There are a number of different ways to calculate the PCA numerically. A description of the various methods is given in section 3.3. As mentioned earlier chemometrics is often used for developing regression models. The development of regression models has a long history in statistics and of course many methods are available. Some of these methods are discussed in section 3.4. The regression method called projection to latent

structures or partial least squares (PLS) is a special regression method used in chemometrics for highly collinear data. This method is described in section 3.5.

In order to handle batch data by methods developed for continuous data the data matrix can be unfolded as described in section 3.6. Fault diagnosis methods based on statistical process monitoring methodologies is described in section 3.7 where methods utilising PCA as modelling principle are given. Methods using PLS instead also exist and are quite similar to the PCA methods. Fault diagnosis using PLS is described in section 3.8.

Finally in section 3.9 summary comments are given and an outline of desirable future directions within the area of process chemometrics for fault diagnosis.

3.1 Principal Component Analysis

Chemical processes often have numerous sensors hooked into the process plant. These measurements are usually not independent since they describe the same events therefore the number of variables can be reduced without losing information.

Principal component analysis (PCA) is a method, which by linear transformations of the variables obtains new uncorrelated variable [Jackson, 1991]. These new latent variables are called principal components (pc's). Often a small number of pc's can be used to describe most of the variation of the original variables. The reduction of the number of dimensions is illustrated in figure 3.1 where a simple data set consisting of only two variables are plotted against each other. It can be seen that the variables x_1 and x_2 are highly correlated. The 95% confidence region indicated in figure 3.1 shows where the major part of the points in the data set should be situated. When the data *do* conform to the model the data points should be inside and only 5% of the points should be outside this region. When the data are correlated PCA can be performed where first the direction with the highest degree of variability is identified. This is the first principal component. Additional principal components may be identified one at a time. The subsequent principal components must be perpendicular (orthogonal) to the previous principal components and point in the direction of the largest variation. The principal components can be used as a new coordinate system. This coordinate system is illustrated in figure 3.2. In this figure it can be seen that when the data are conforming to the model (are inside the confidence region) the data only spans one dimension in this case. The second dimension can be discarded with little error. This example illustrates the main feature of PCA; it transforms data into a few orthogonal principal components. The effect of using PCA is larger when higher dimensional data are analysed. E.g. spectroscopical data having 100–1000 frequencies can be reduced into perhaps 5 to 10 principal components [Martens and Næs, 1989].

The notation used here is mainly from [Jackson, 1991]. Appendix A and B gives an introduction to the notation used in the areas of linear algebra and statistics respectively.

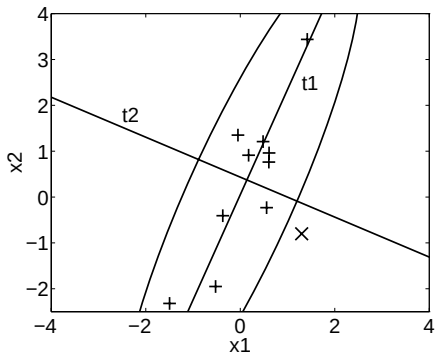


Figure 3.1. Original data set with correlated data. The ellipse shows the 95% confidence interval for the data under normality assumptions. The axes in the ellipse define the principal components. The point marked with a cross seems not to conform to the other data since it is outside the confidence region.

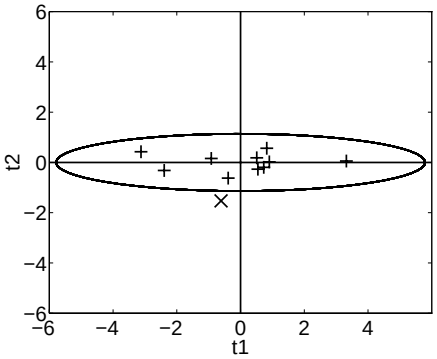


Figure 3.2. Transformed data set with uncorrelated data. It can be seen that the data set mainly spans one dimension. The data point marked with a cross is outside the confidence region and indicates that a second dimension is used for this point.

3.1.1 Eigenvectors and Eigenvalues

Corresponding to the data matrix $\mathbf{X}$ $(I \times J)$ there is a covariance matrix $\mathbf{S}$ $(J \times J)$. Normally I is the number of experiments (contained in rows) and J is the number is variables (contained in columns). The covariance matrix can be decomposed in the following way

$$\mathbf{S} = \mathbf{U}\mathbf{L}\mathbf{U}^\top, \tag{3.1}$$

where the columns of $\mathbf{U}$ are the *orthonormal eigenvectors* of $\mathbf{S}$ and $\mathbf{L} = \text{diag}(l_1, l_2, \ldots, l_J)$ contains the *eigenvalues* in descending order. The eigenvectors are called *loadings*.
New variables can be defined on basis of $\mathbf{U}$:

$$\mathbf{z} = \mathbf{U}^\top(\mathbf{x} - \bar{\mathbf{x}}), \tag{3.2}$$

where $\mathbf{x}$ is a column vector of measurements and $\bar{\mathbf{x}}$ is the mean vector of $\mathbf{X}$. These new variables will be called principal components (pc's). Individual transformed observations will be called z-scores. The main advantage of this transformation is that the individual z-scores will be uncorrelated.
The jth score is

$$z_j = \mathbf{u}_j^\top(\mathbf{x} - \bar{\mathbf{x}}) \tag{3.3}$$

It will have zero mean and variance l_j.

New variables can be made on basis of $\mathbf{U}$:

$$\mathbf{Z} = \mathbf{XU}. \tag{3.4}$$

Another way of writing this relationship is

$$\mathbf{X} = \mathbf{ZU}^\top \tag{3.5}$$

$$= \tilde{\mathbf{Z}}\tilde{\mathbf{U}}^\top + \mathbf{E}, \tag{3.6}$$

$$\approx \tilde{\mathbf{Z}}\tilde{\mathbf{U}}^\top \tag{3.7}$$

where $\tilde{\mathbf{Z}}$ and $\tilde{\mathbf{U}}$ are the first A columns of $\mathbf{Z}$ and $\mathbf{U}$ respectively. A is the number of retained principal components. $\mathbf{E}$ contains unmodelled part of the data. The tildes (e.g. $\tilde{\mathbf{U}}$) will only be used when necessary and it will be assumed, hereafter, that the vectors and matrices only contain the necessary number of elements as determined by the model.

The scores are then calculated by

$$\mathbf{Z} = \mathbf{X}\tilde{\mathbf{U}} \tag{3.8}$$

Since the eigenvalues are in descending order the error made when using (3.7) instead of (3.5) is usually small. The error will of course depend on the data and the choice of A. This choice will be dealt with in section 3.2.

3.1.2 Interpretation of the Principal Components

The pc's will almost always lack any direct physical meaning because they are composed of usually very different variables, but by examining the matrices involved some physical meaning and insight can be extracted.

The scores are uncorrelated. It means that the different scores describe totally different events *in the data*. One can only hope that the scores describe totally different events *in the process*.

For each row in $\mathbf{X}$ there is a corresponding row in $\mathbf{Z}$. The data points in the different rows of $\mathbf{X}$ usually come from different experiments. We then see that the different rows of $\mathbf{Z}$ describe the difference between the experiments in a reduced space.

The eigenvectors $\mathbf{U}$ describe the *rotation* of the original data, i.e. the *direction* of the transformation. Each element in, e.g., the first column of $\mathbf{U}$ indicate the weight the corresponding variable has on the first pc. Some weights are low, indicating that the variables has little weight on the pc. By examining the values some relationship between the measured variables and the pc's can usually be extracted. The first pc can usually be assigned as an overall indicator of what is going on (e.g. the reaction rate). The succeeding pc's may be interpretable too, but that is normally true only for the first 2 or 3 pc's.

The interpretation of the components depends very much on the application and the physical knowledge of the process. An Example of interpretable pc's can be found in [Jackson, 1991, page 69–70].

3.1.3 Scaling

PCA is scale dependent. When dealing with a covariance matrix based on variables with different units or variables with large differences in scale, the largest eigenvalues will reflect the variables with the largest magnitude. Since it is not generally true that variables of large magnitude are the most important, scaling of the variables is required.

Scaling can be applied to the raw data before PCA analysis or it can be applied to the pc's. It is important that the correct scaling of the raw variables is chosen and that a suitable scaling of the pc's is chosen to give optimal presentation of the result [Jackson, 1991]. The two ways of scaling have different effect and is described below.

3.1.3.1 Scaling of the Principal Components

The **u**-vectors can be scaled in the following two popular ways

$$\mathbf{v}_i = \sqrt{l_i}\,\mathbf{u}_i \qquad \mathbf{V} = \mathbf{U}\mathbf{L}^{1/2} \tag{3.9}$$

$$\mathbf{w}_i = \mathbf{u}_i/\sqrt{l_i} \qquad \mathbf{W} = \mathbf{U}\mathbf{L}^{-1/2} \tag{3.10}$$

This leads to new scores. One is defined by

$$\sqrt{l_i}\; z_i = \mathbf{v}_i^\top(\mathbf{x} - \bar{\mathbf{x}}) \tag{3.11}$$

which has the same units as the original data. The variance is the square of the eigenvalues.
Using $\mathbf{W}$ the y-scores are formed

$$y_i = \mathbf{w}_i^\top(\mathbf{x} - \bar{\mathbf{x}}) \quad \text{or} \quad \mathbf{y} = \mathbf{W}^\top(\mathbf{x} - \bar{\mathbf{x}}). \tag{3.12}$$

They have all unit variance which is a nice property for testing purposes.
The matrices defined in (3.9) and (3.10) have the following properties.

$$\mathbf{V}^\top\mathbf{V} = \mathbf{L} \qquad \mathbf{V}\mathbf{V}^\top = \mathbf{S} \qquad \mathbf{V}^\top\mathbf{S}\mathbf{V} = \mathbf{L}^2 \tag{3.13}$$

$$\mathbf{W}^\top\mathbf{W} = \mathbf{L}^{-1} \qquad \mathbf{W}\mathbf{W}^\top = \mathbf{S}^{-1} \qquad \mathbf{W}^\top\mathbf{S}\mathbf{W} = \mathbf{I} \tag{3.14}$$

3.1.3.2 Scaling of the Variables

PCA is sensitive to the scaling of the variables. Using PCA the maximal variation in some direction is found. If the scale of the variables in that direction is changed, say diminished, the variation in that direction will appear to have become smaller. In this section different scaling techniques of data will be demonstrated.

It can be necessary to scale data in three cases:

1. The original variables are in different units that describe the same kind of measurements. e.g., centimetres and kilometres are both used to describe

length. If a variable is measured in inches it will determine the direction of maximal variation much more than if it was measured in kilometres, simply because the number of inches will be larger than the number of kilometres.

2. The original variables are in different units that describe different kinds of measurements. The same argument as before holds if we compare inches with kilograms. It is difficult to define in a meaningful way if one inch is larger than one kilogramme. The interpretation of the units of the elements of the covariance matrix and the resulting pc's also becomes difficult in this case.

3. The original variables are in the same units, but have very different ranges.

The problems listed describe the problems of PCA: Some *physical quantities* are described with *numbers*. PCA is a statistical method that works with these numbers expecting that some physical phenomena can be described. If the numbers are not comparable the effect of different units and magnitude of variables can render the analysis worthless.

The data can be scaled in different ways. One way is to scale the variables by some known maximal value or upper control limit. Another way is to scale the variables such that they have variance 1.0. Usually the latter is accompanied by a centring of the variables and then the scaling is called *autoscaling*. The covariance matrix obtained on autoscaled data is equal to the correlation matrix of the original data.

These scaling procedures will make sure that the variables will not be of totally different magnitude. The autoscaling procedure is assumed used on all data sets in this thesis when PCA and PLS are to be employed.

3.2 Number of Principal Components

A decision has to be made: How many pc's are necessary to explain the variation in the original variables; how is the optimal value of A defined in section 3.1.1 determined. This can be determined by inspecting the eigenvalues, by performing statistical tests or by cross validation, which will be described in the following sections. This discussion is also relevant for the PLS models that will be described in section 3.5.

3.2.1 Covariance matrix of less than full rank

If the rank of $\mathbf{S}$ is less than J, which means that one or more of l_i is zero, one or more linear relationships exist between the variables. This implies that variables can be removed without losing *any* information.

3.2.2 Eigenvalues Almost Equal

If some of the eigenvalues are almost equal then the order of the corresponding eigenvectors will be undefined. The physical interpretation of this is that these eigenvectors describe noise. This will usually be the case for the eigenvectors associated with the last, small eigenvalues.

3.2.3 Statistical Tests.

A number of tests and rules of thumb exist for determining the significant eigenvectors and eigenvalues when they have been determined from the covariance matrix [Jackson, 1991].

When the correlation matrix is used—which is the case when data has been autoscaled—the eigenvalues do not depend on the covariances but of their ratios. This makes the tests weak mainly because the distributions involved in the tests become difficult to find and (bad) approximations has to be used.

One test that can be used as a stopping rule is a test for the last $(J - A)$ eigenvalues being equal

$$\mathrm{H}_0 : \; l_{A+1} = l_{A+2} = \cdots = l_J. \tag{3.15}$$

To understand why this is a good test one has to inspect the associated eigenvectors. If some of the eigenvalues are equal there will not be *one* eigenvector corresponding to *one* eigenvalue, but instead there will be several eigenvectors of equal importance. These eigenvectors will span a space. That means that they are not determining a direction. Thus the pc's are undetermined.

It is possible, but rare, that some of the first eigenvalues are equal. For this case there are no tests when the correlation matrix is concerned.
To test the hypothesis (3.15) calculate

$$c \left\{ \sum_{j=A+1}^{J} \ln(l_j) \; + \; (J - A) \ln \left(\sum_{j=A+1}^{J} \frac{l_j}{J - A} \right) \right\}, \tag{3.16}$$

where $c = I - (1/6)(2J + 5) - (2/3)A$. The expression in equation (3.16) will have a χ^2 distribution only if $l_1, \ldots, l_A$ are large relative to the remaining eigenvalues.

Let $\mathbf{C} = \mathbf{W}_{A+1:J}\mathbf{W}_{A+1:J}^{\top}$ and $\bar{l} = (\sum_{j=A+1}^{J} l_j)/(J-A)$. Then calculate

$$F = 2(J-A-1)\,\bar{l}\sum_{j=1}^{J} c_{jj}^2 \tag{3.17}$$

$$G = (J-A)\sum_{i=1}^{J}\sum_{j=1}^{J} c_{ij}^2 r_{ij}^2 \tag{3.18}$$

$$H = \sum_{i=1}^{J}\sum_{j=1}^{J} c_{ii}c_{jj}r_{ij}, \tag{3.19}$$

where r_{ij} is an element of the correlation matrix $\mathbf{R}$ of the data. The degrees of freedom in the χ^2 distribution are

$$\frac{1}{2}(J-A-1)(J-A+2) - \frac{1}{J-A}(F-G+H). \tag{3.20}$$

3.2.4 Explanation of Variance

The sum of the variance of the variables is

$$s_1^2 + s_2^2 + \cdots + s_J^2 = \operatorname{tr}(\mathbf{S}) = \operatorname{tr}(\mathbf{L}). \tag{3.21}$$

The eigenvalues l_j are the variances of the pc's. The sum of variances is the same for the original variables and the pc's. The number

$$d_j = l_j/\operatorname{tr}(\mathbf{S}) \tag{3.22}$$

thus indicates the fraction of the variance in the original data that the jth pc explains. The first A pc's explain $\sum_{j=1}^{A} d_j$.

It is desirable that the pc's explain a large amount of variation in $\mathbf{X}$, and if all pc's are retained this is accomplished. Using all the pc's is usually not a good idea

If the explained fraction is too small and one can not safely include more pc's in the model; maybe PCA is not possible on the data.

Simply selecting the number of components based on the explained variance may give a large number of pc's in the model since often a fairly large number of components are needed to describe e.g. 95% or 99% of variation. By predetermining such a percentage before the analysis of the data is performed too much noise may be included in the model. The individual contributions of the pc's is also important and will be described next.

3.2.5 Inspection of Eigenvalues

The eigenvalues can be inspected manually either in their numerical form or graphically. When a drastic drop in the eigenvalues is detected is it safe to cut off the remaining eigenvalues and eigenvectors.

Two examples of eigenvalues using simulated data are presented in figures 3.3 and 3.4 . In figure 3.3 an example is shown where a PCA model would require about 8 components before the eigenvalues are small enough to conclude that adding extra components will not add extra information to the model, but will possibly only add noise. The curve on figure 3.4 has a more clear bend at the point where 4 components are included in the model although there seen to be some information in the 5th and 6th components as well.

On real data the break in the eigenvalues are not always sharp and it becomes more difficult to determine an optimal number of pc's. Another thing that has to be considered is the application of the PCA model. Sometimes it may be beneficial to incorporate more or less pc's in the model. The eigenvalue structure can thus only be used as a *guide* for selecting an optimal number of pc's.

3.2.6 Cross-validation.

Another method for selection the number of components to keep in the model is cross-validation. By finding eigenvectors and eigenvalues for one data set and validating it by means of another it is possible to ascertain how many eigenvectors are sufficient to explain the variation in $\mathbf{X}$. The validation method must depend on the later use of the model.

More methods for the selection the number of components can be found in [Jackson, 1991; Höskuldsson, 1996; Hansen, 1996].

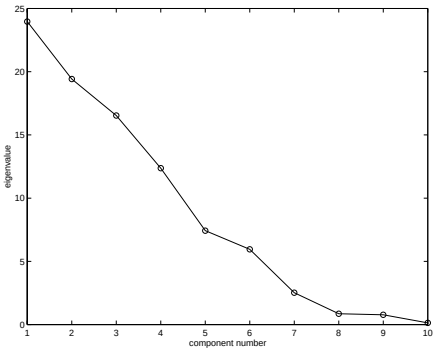


Figure 3.3. Example of eigenvalues. The eigenvalues decrease slowly, but there is no distinct point where the change in eigenvalues are large.

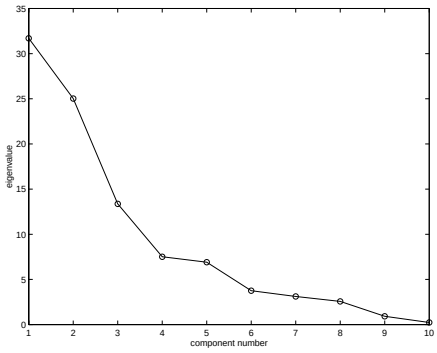


Figure 3.4. Example of eigenvalues. The eigenvalues decrease faster in this figure. There is a large difference between eigenvalue 5 and 6. Thus an initial estimate is 5 components for a PCA model.

3.3 Calculating the PCA in detail

Below an in depth description of the numerics involved in the calculation of the PCA is given in order to give an accurate method for performing the involved calculations.

Data generally come in two types of matrices; they are either long and narrow or they are short and wide. These two types of matrices can of course be handled in the same way, but this is not optimal since it may require a substantial amount of computer memory and computation time if the matrices are not handled correctly. Furthermore incorrect handling of the data may lead to loss of precision.

The basic property of PCA is that scores are extracted using the following maximisation problem. The jth score is the $\mathbf{z}$ that solves

$$\max \|\mathbf{z}_j\|^2 = \max \mathbf{z}_j^\top \mathbf{z}_j = \max_{\|\mathbf{u}_j\|=1} \mathbf{u}_j^\top \mathbf{X}^\top \mathbf{X} \mathbf{u}_j. \tag{3.23}$$

The following scores are found by reducing the rank of $\mathbf{X}$ using the expression

$$\mathbf{X}_{new} = \mathbf{X}_{old} - \mathbf{z}_j \mathbf{u}_j^\top. \tag{3.24}$$

This assures that the loadings and scores become orthogonal. The new $\mathbf{X}$ is then used in the maximisation.

3.3.1 Eigenvalue Method

In this section it will be assumed that $\mathbf{X}$ $(I \times J)$ is centred (see section B.1.4). The unbiased positive semi-definite covariance matrix of $\mathbf{X}$ defined by

$$\mathbf{S} = \frac{1}{I-1} \mathbf{X}^\top \mathbf{X} \tag{3.25}$$

can always be decomposed into real eigenvalues $\mathbf{L}$ and eigenvectors $\mathbf{U}$:

$$\mathbf{S} = \mathbf{U} \mathbf{L} \mathbf{U}^\top, \tag{3.26}$$

where $\mathbf{L}$ is $(r \times r)$ and $\mathbf{U}$ is $(J \times r)$. r is the rank of $\mathbf{S}$.
A matrix $\mathbf{S}_*$ is defined

$$\mathbf{S}_* = \frac{1}{I-1} \mathbf{X} \mathbf{X}^\top. \tag{3.27}$$

This matrix has the same eigenvalues as $\mathbf{S}$ [Basilevsky, 1983]. The matrix of eigenvectors is $\mathbf{U}_*$ $(I \times r)$. This can be related to $\mathbf{U}$ by the following expression

$$\mathbf{U} = \mathbf{X}^\top \mathbf{U}_* \mathbf{L}^{-1/2} (I-1)^{-1/2} \tag{3.28}$$

Using equations (3.27) and (3.28) it can be seen that these can be stated in an even shorter way. Let the product matrix

$$\mathbf{S}_{**} = \mathbf{X} \mathbf{X}^\top \tag{3.29}$$

have the eigenvalues $\mathbf{L}_{**}$. The eigenvector matrix is $\mathbf{U}_*$. The eigenvectors of $\mathbf{S}$ can then be calculated by

$$\mathbf{U} = \mathbf{X}^\top \mathbf{U}_* {\mathbf{L}_{**}}^{-1/2} \tag{3.30}$$

$\mathbf{U}$ can be reduced to an even smaller size by only using the first few (A) significant columns of $\mathbf{U}_*$.

If only a few columns of $\mathbf{U}$ are needed this method will give a reasonable result. If all (or almost all) columns are needed, too much information will be lost because of rounding errors in the calculation of $\mathbf{S}_*$ or $\mathbf{S}_{**}$ [Golub and van Loan, 1991].

When $\mathbf{X}$ is long and narrow ($I \gg J$) $\mathbf{S}$ will be much smaller than $\mathbf{S}_*$. When $\mathbf{X}$ is short and wide ($J \gg I$) $\mathbf{S}_*$ is smaller than $\mathbf{S}$.

3.3.2 NIPALS

The numerical method normally used in chemometrics is the NIPALS algorithm (Nonlinear Iterative Partial Least Squares). The description and the notation used here will be the one used in the literature elsewhere (e.g. [Wold *et al.*, 1987a; Nomikos and MacGregor, 1995; Wold *et al.*, 1987b; Kresta *et al.*, 1991]).

3.3.2.1 Approximation of the Data Matrix

The data matrix $\mathbf{X}$ ($I \times J$) is by the NIPALS algorithm approximated by a sum of outer products of some vectors $\mathbf{t}$ and $\mathbf{p}$.

$$\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2 + \cdots + \mathbf{X}_A + \mathbf{E} \tag{3.31}$$

$$= \mathbf{t}_1\mathbf{p}_1^\top + \mathbf{t}_2\mathbf{p}_2^\top + \cdots + \mathbf{t}_a\mathbf{p}_a^\top + \ldots \mathbf{t}_A\mathbf{p}_A^\top + \mathbf{E} \tag{3.32}$$

$$= \mathbf{T}\mathbf{P}^\top + \mathbf{E}. \tag{3.33}$$

The matrices $\mathbf{X}_1, \ldots, \mathbf{X}_A$ are rank 1 matrices and $\mathbf{E}$ is a residual matrix. $\mathbf{T}$ ($I \times A$) and $\mathbf{P}$ ($J \times A$) is calculated in such a way that $\mathbf{E}$ is as small as possible in a least squares sense. A is chosen so $\mathbf{E}$ does not contain any significant process information. The vectors $\mathbf{t}_1, \ldots, \mathbf{t}_A$ and $\mathbf{p}_1, \ldots, \mathbf{p}_A$ are by the definition above orthogonal. In order to facilitate an interpretation of the decomposition the vectors $\mathbf{p}_1, \ldots, \mathbf{p}_A$ are chosen to be orthonormal. $\mathbf{P}$ in this section is equal to $\mathbf{U}$ mentioned in the previous section.

3.3.2.2 Interpretation

The decomposition into $\mathbf{t}$ and $\mathbf{p}$ vectors is shown in figure 3.5. Here it can be seen that $\mathbf{p}$ is related to the columns of $\mathbf{X}$, and that $\mathbf{t}$ is related to the different rows.

$\mathbf{p}_a$ is called a loading vector. It summarises the variation of the measurement variables around their mean trajectories at every discrete point in time corresponding to the time span of the normal data set.

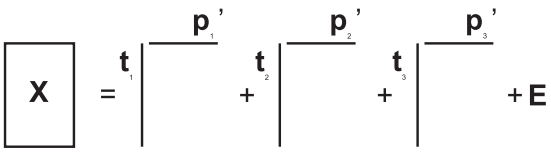


Figure 3.5. Decomposition of $\mathbf{X}$ when using PCA. The vectors $\mathbf{p}_a$ are orthonormal.

$\mathbf{t}_a$ is a score vector. It gives an indication of the variation of a single batch with respect to the total data set throughout the whole batch duration.

3.3.2.3 The NIPALS Algorithm

The NIPALS algorithm exists in many variations. The one presented here is from [Nomikos and MacGregor, 1995]. The algorithm is similar to the power method used to obtain eigenvectors and eigenvalues [Lorber *et al.*, 1987] (for an explanation of the power method see [Golub and van Loan, 1991, page 351].

The steps of the algorithm can be seen in table 3.1. The algorithm can be summarised if we observe the main iteration loop (step 5–7). After convergence a constant c_a equal to the length of $\mathbf{p}_a$ in step 5 can be defined:

$$c_a = \underbrace{\|\mathbf{p}_a\|}_{\text{from step 5}} = \|\mathbf{E}^\top \mathbf{t}_a\| = \mathbf{t}_a^\top \mathbf{t}_a. \tag{3.34}$$

If the expression for $\mathbf{p}_a$ in step 5 is expanded using the expressions in step 6 and 7 we get

$$\mathbf{p}_a = \mathbf{E}^\top \mathbf{E}\, \mathbf{p}_a / c_a \qquad \Leftrightarrow \qquad \mathbf{E}^\top \mathbf{E}\, \mathbf{p}_a = c_a\, \mathbf{p}_a. \tag{3.35}$$

It can be seen (compare with equation (A.7)) that c_a is the ath eigenvalue of $\mathbf{E}^\top \mathbf{E}$ (equal to the square of the ath singular value of $\mathbf{E}$) and that $\mathbf{p}_a$ is the eigenvector.

3.3.2.4 Properties

The NIPALS algorithm has the same properties as the power method.

1. It only works when $c_a / c_{a+1} > 1$
2. It finds the eigenvalues in descending order
3. The algorithm can be stopped when the desired number of eigenvalues and eigenvectors have been obtained.
4. It is very simple
5. The accuracy, however, is very low because of accumulation of rounding errors

1	Scale $\mathbf{X}$ to standard units
2	$a = 1$
3	Choose the column of $\mathbf{X}$ which has the greatest variance as $\mathbf{t}_a$
4	$\mathbf{E} = \mathbf{X}$
5	$\mathbf{p}_a = \mathbf{E}^\top \mathbf{t}_a$
6	$\mathbf{p}_a = \mathbf{p}_a / \|\mathbf{p}_a\|$
7	$\mathbf{t}_a = \mathbf{E}\mathbf{p}_a$
8	If $\mathbf{t}_a$ has converged goto 9 else goto 5
9	If $a = A$ then stop
10	$\mathbf{E} = \mathbf{E} - \mathbf{t}_a \mathbf{p}_a^\top$
11	$a = a + 1$
12	$\mathbf{t}_a = \mathbf{t}_{a-1}$
13	goto 5

Table 3.1. The NIPALS algorithm

Property 1 does hardly ever constitute a problem when the algorithm is used on real data and is only a theoretical hindrance, but the fraction c_a / c_{a+1} does have an influence on the convergence speed. When it is close to one the algorithm converges very slowly. This is also a reason why the method should not be used to extract all components. Property 2, 3 and 4 explains why NIPALS is so popular. Property 5 is a reason why NIPALS should be used only when the first few eigenvalues and eigenvectors are wanted and even then a decreasing accuracy can be noticed.

3.3.3 SVD

The economy size singular value decomposition of $\mathbf{X}$ $(I \times J)$ is (see section A.4)

$$\mathbf{X} = \mathbf{U}\mathbf{M}\mathbf{V}^\top, \tag{3.36}$$

where both $\mathbf{V}$ and $\mathbf{U}$ are orthonormal and $\mathbf{M}$ is a diagonal matrix. Note that $\mathbf{U}$ from the SVD is not equal to the $\mathbf{U}$ matrix obtained from the eigenvalue decomposition.

By comparing with equation (3.33) it can be seen that SVD is related to NIPALS, where $\mathbf{T} = \mathbf{U}\mathbf{M}$ and $\mathbf{P} = \mathbf{V}$.

The eigenvectors of $\mathbf{S}$ are equal to $\mathbf{P} = \mathbf{V}$ and the eigenvalues $\mathbf{L}$ can be found from $\mathbf{T}$ or $\mathbf{M}$

$$\mathbf{L} = \mathrm{diag}(\mathbf{t}_1^\top \mathbf{t}_1, \mathbf{t}_2^\top \mathbf{t}_2, \ldots, \mathbf{t}_A^\top \mathbf{t}_A)/(I-1) \tag{3.37}$$

$$= \mathbf{M}^2/(I-1) \tag{3.38}$$

The singular value decomposition is a very robust method for decomposing matrices. It finds the matrices $\mathbf{U}$, $\mathbf{M}$ and $\mathbf{V}$ simultaneously and distributes the errors of the calculation [Golub and van Loan, 1991].

SVD-algorithms exist that only find the first principal components. Usually any standard routine, which finds all pc's, will be just as fast unless $\mathbf{X}$ possesses special properties (e.g. sparsity or is very large).

The three methods described can all be used to obtain $\mathbf{U}$ and $\mathbf{L}$. It should be noticed that none of the methods give an unique definition of the sign of the eigenvectors. Deviations between the methods can therefore exist and any mixing should be done carefully.

The SVD method is without question the most robust but is also the most complicated. Routines for the SVD can easily be at more than 10,000 lines of C-code in order to provide robustness, but smaller routines do exist [Press *et al.*, 1992]. Although the SVD may take longer time than the other methods, the difference is almost unnoticeable.

The NIPALS algorithm is very simple which is the reason for its popularity. Finding the eigenvalues and eigenvectors is an off-line post treatment of the data and there is really not any reason for keeping the calculation simple when accuracy is sacrificed.

3.4 Linear Regression

3.4.1 Introduction

Experimenters often want to determine a model for the effect on some dependent variables when some independent variables are changed. The independent variables $\mathbf{x}$ are known as inputs and the corresponding dependent variables $\mathbf{y}$ are called outputs. Often $\mathbf{x}$ is easily obtained and it is desirable to find a way to estimate $\mathbf{y}$ based on $\mathbf{x}$.

In the following sections we will study the linear relation

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}. \tag{3.39}$$

Here $\mathbf{Y}$ and $\mathbf{X}$ are matrices of measurements obtained from experiments. Both contain realisations of variables as rows. $\mathbf{B}$ is the matrix of coefficients that determine the model. $\mathbf{E}$ represents unmodelled noise.

It is the purpose of this section to describe ways to determine $\mathbf{B}$. The solution method depend on the number of experiments and the number of variables in $\mathbf{X}$ and $\mathbf{Y}$. Furthermore will the degree of linear dependence between variables in $\mathbf{X}$ and $\mathbf{Y}$ have a large influence on the methods that are feasible.

In the following sections various classical estimation methods for linear models will be described. Multivariate linear regression is defined in section 3.4.2. Ordinary least squares methods for estimating $\mathbf{B}$ are described in section 3.4.3. The corresponding maximum likelihood methods for solving the estimation problem are described in section 3.4.4.

When the variables are collinear the ordinary least squares and maximum likelihood estimation methods are numerically highly sensitive. One of the

ways to improve the estimation method in this case is to use ridge regression described in section 3.4.5. More robust methods are principal component regression (PCR) described in section 3.4.6 and projection to latent structures (PLS) described in section 3.5.

3.4.2 Multivariate Linear Regression

Consider the linear relation

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}, \tag{3.39}$$

where $\mathbf{Y}$ $(I \times M)$ contains the response or output variables and $\mathbf{X}$ $(I \times J)$ contains the design or input variables. I is the number of experiments, M is the number of output (measured) variables and J is the number input variables. $\mathbf{Y}$ is always based on measurements, where as $\mathbf{X}$ can be measured or it can be known *exactly*. $\mathbf{B}$ $(J \times M)$ is the parameter matrix that determine the model. $\mathbf{E}$ $(I \times M)$ represents unmodelled noise.

It is the purpose of multiple linear regression (MLR) to estimate $\mathbf{B}$ from $\mathbf{X}$ and $\mathbf{Y}$ matrices based on many measurements in such a way that it later, based on a new $\mathbf{x}$, may be possible to estimate $\mathbf{y}$[1].

3.4.3 Ordinary Least Squares

If there exists a linear relationship between the $\mathbf{X}$ and $\mathbf{Y}$ variables in (3.39) the ideal model must clearly be the one where $\mathbf{E}$ is equal to the zero matrix. It is the same as solving the equation $\mathbf{Y} = \mathbf{XB}$. Solving this three cases must be considered:

$N = K$	If $\mathbf{X}^{-1}$ exists there is one solution $\mathbf{B} = \mathbf{X}^{-1}\mathbf{Y}$. This solution is exact in the sense that $\mathbf{E} = \mathbf{0}$.
$N > K$	No solutions (normally).
$N < K$	Either no or infinitely many solutions exist.

In the case where the observations are a sum of the true underlying value of the measured quantity and some noise, it not likely that a $\mathbf{B}$ can be found that satisfy $\mathbf{Y} = \mathbf{XB}$ not even in the case where the underlying system actually *is* linear. It is then the task of the estimator to find a $\mathbf{B}$ that makes $\mathbf{E}$ small in some sense. Ordinary least squares (OLS) methods measures the size of the matrix $\mathbf{E}$ using the 2-norm. This will be utilised in the following sections.

3.4.3.1 Multiple Linear Regression

The principle for the least squares solution of the linear regression problem is illustrated in figure 3.6. The case where there is only one $\mathbf{y}$-variable and several $\mathbf{x}$-variables is called *multiple linear regression*. This case has been chosen

[1] In statistics it is customary to write estimated quantities with a "hat", e.g. $\hat{b}$. Because everything in this thesis will be based on estimates the "hats" are only written when absolutely necessary in order to avoid an excessive use of symbols.

because it easier to understand (and display) and the generalisation to the case where there are more $\mathbf{y}$-variables is straightforward because the structure of $\mathbf{Y}$ does not enter directly in the calculations.

It is assumed that $\mathbf{X}$ $(I \times J)$ is known exactly and that $\mathbf{y}$ $(I \times 1)$ depends linearly on $\mathbf{X}$. $\mathbf{y}$ contains noise. The relationship $\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$ can for each measurement be written

$$y_n = \mathbf{x}_{(n)}^{\top}\mathbf{b} + e_n, \tag{3.40}$$

where $\mathbf{x}_{(n)}$ is the nth row of $\mathbf{X}$. The noise $\mathbf{e}$ has expectation $\mathbf{0}$ and variance $\sigma^2\mathbf{I}$, but there is no restriction on the distribution of $\mathbf{e}$ [Mardia *et al.*, 1995]. It is desirable to find a $\mathbf{b}$ that makes the residual $\mathbf{e}$ small in some sense. The residuals e_n are indicated on figure 3.6 as the vertical lines between the line and the data points.

The problem can be formulated more precisely [Golub and van Loan, 1991]

$$\mathbf{b} = \arg \min_{\mathbf{b} \in \mathbb{R}^K} \|\mathbf{X}\mathbf{b} - \mathbf{y}\|_p. \tag{3.41}$$

When p is equal to 2 we have a *least squares* problem. The reason $p = 2$ is preferred can be summarised as

- Equal weight on positive and negative deviations.
- High weight on large residuals, because the deviations are squared.
- The function defined by $\phi(\mathbf{b}) = \frac{1}{2}\|\mathbf{X}\mathbf{b} - \mathbf{y}\|_2^2$ is a differentiable function of $\mathbf{b}$.

The solution to the least squares problem is the $\mathbf{b}$ that minimises ϕ.

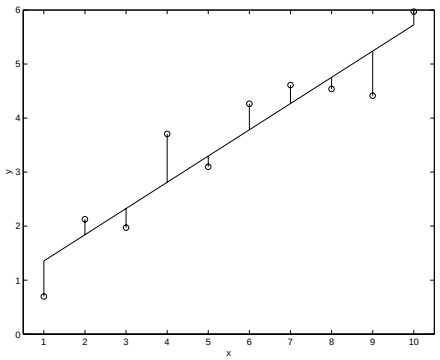


Figure 3.6. Principle of the least squares method in the case where there are only one $\mathbf{x}$-variable and one $\mathbf{y}$-variable.

First ϕ is expanded

$$\begin{aligned}\phi(\mathbf{b}) &= \frac{1}{2}\|\mathbf{X}\mathbf{b}-\mathbf{y}\|_2^2 \\ &= \frac{1}{2}(\mathbf{X}\mathbf{b}-\mathbf{y})^\top(\mathbf{X}\mathbf{b}-\mathbf{y}) \\ &= \frac{1}{2}\left(\mathbf{b}^\top\mathbf{X}^\top\mathbf{X}\mathbf{b}-2\mathbf{b}^\top\mathbf{X}^\top\mathbf{y}\right)\end{aligned}$$

then differentiated twice (see section A.7).

$$\frac{\partial\phi(\mathbf{b})}{\partial\mathbf{b}} = \mathbf{X}^\top\mathbf{X}\mathbf{b}-\mathbf{X}^\top\mathbf{y}. \tag{3.42}$$

$$\frac{\partial^2\phi(\mathbf{b})}{\partial\mathbf{b}^2} = \mathbf{X}^\top\mathbf{X}. \tag{3.43}$$

If $\mathbf{X}$ has *full column rank* a unique solution can be found. A necessary condition for an extremum of $\phi(\mathbf{b})$ is that equation (3.42) is set equal to the zero matrix. Since (3.43) is *positive definite* that solution *is* the minimum. Thus, the solution $\mathbf{b}$ to the least squares problem is the one that solves the *normal equations*

$$\mathbf{X}^\top\mathbf{X}\mathbf{b} = \mathbf{X}^\top\mathbf{y}. \tag{3.44}$$

If $\mathbf{X}$ does not have full column rank (this is always true if $I < J$) the normal equations does give a unique solution. It is then necessary to introduce restrictions on $\mathbf{b}$ [Björck, 1994]

$$\min_{\mathbf{b}\in\mathbb{S}}\|\mathbf{b}\|_2, \quad \mathbb{S} = \{\mathbf{b}\in\mathbb{R}^J \mid \mathbf{b} = \arg\min\|\mathbf{X}\mathbf{b}-\mathbf{y}\|_2\}. \tag{3.45}$$

If $\mathbf{X}$ has *full row rank* then the solution can be found from the *normal equations of the second kind*

$$\mathbf{X}\mathbf{X}^\top\mathbf{z} = \mathbf{y}, \qquad \mathbf{b} = \mathbf{X}^\top\mathbf{z} \tag{3.46}$$

that have the solution $\mathbf{b} = \mathbf{X}^\top(\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{y}$.

Irrespective of whether $\mathbf{X}$ has full column rank or full row rank the least squares problem (3.45) can be solved using the *pseudo inverse* of $\mathbf{X}$ [Golub and van Loan, 1991]. If the SVD of $\mathbf{X}$ is $\mathbf{X} = \mathbf{U}\mathbf{M}\mathbf{V}^\top$ (see section A.4) and $r = \text{rank}(\mathbf{X})$, we can write

$$\begin{aligned}\|\mathbf{X}\mathbf{b}-\mathbf{y}\|_2^2 &= \|\mathbf{U}^\top\mathbf{X}\mathbf{b}-\mathbf{U}^\top\mathbf{y}\|_2^2 \\ &= \|(\mathbf{U}^\top\mathbf{X}\mathbf{V})(\mathbf{V}^\top\mathbf{b})-\mathbf{U}^\top\mathbf{y}\|_2^2 \\ &= \|\mathbf{M}\boldsymbol{\alpha}-\mathbf{U}^\top\mathbf{y}\|_2^2 \\ &= \sum_{i=1}^{r}(\sigma_i\alpha_i-\mathbf{u}_i^\top\mathbf{y})^2 + \sum_{i=r+1}^{I}(\mathbf{u}_i^\top\mathbf{y})^2,\end{aligned} \tag{3.47}$$

where $\boldsymbol{\alpha} = \mathbf{V}^\top \mathbf{b}$, and we have used that $\|\mathbf{U}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$ whenever $\mathbf{U}$ is orthonormal and that both $\mathbf{U}$ and $\mathbf{V}$ are orthonormal. It is seen that in order to minimise (3.47) we must have $\alpha_i = \mathbf{u}_i^\top \mathbf{y}/\sigma_i$ for $i = 1, \ldots, r$. If we for $i = r+1, \ldots, J$ set $\alpha_i = 0$ the resulting $\mathbf{b}$ has minimal 2-norm and is given by

$$\mathbf{b} = \sum_{i=1}^{r} \frac{\mathbf{u}_i^\top \mathbf{y}}{\sigma_i} \mathbf{v}_i \ . \tag{3.48}$$

That means that $\mathbf{b}$ can be calculated using the pseudo inverse of $\mathbf{X}$, see section A.6. The result is simply given by

$$\mathbf{b} = \mathbf{X}^{\#} \mathbf{y}, \tag{3.49}$$

where $\mathbf{X}^{\#}$ is the pseudo inverse of $\mathbf{X}$.

3.4.3.2 Multivariate Linear Regression

In the case where there are more than one $\mathbf{y}$-variable the technique is called multivariate linear regression instead of multiple linear regression. The results from the previous section can be used directly in solving this kind of problem. The $\mathbf{b}$ parameter can be calculated for each column in $\mathbf{Y}$ and stored as columns in $\mathbf{B}$.

The normal equations become

$$\mathbf{B} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{Y}. \tag{3.50}$$

The calculation of $\mathbf{X}^\top \mathbf{X}$ is potentially numerically dangerous because of the unnecessary squaring of the elements of $\mathbf{X}$ and the possibility of overflow. The numerically correct way of calculating $\mathbf{B}$ can be found in [Golub and van Loan, 1991; Dennis and Schnabel, 1983].

The method using pseudo inverse becomes

$$\mathbf{B} = \mathbf{X}^{\#} \mathbf{Y}. \tag{3.51}$$

3.4.4 Maximum Likelihood Estimation

In this section we will take a statistical view on estimating $\mathbf{B}$ in the multivariate linear regression model given by

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{E}, \tag{3.39}$$

The idea of how to find $\mathbf{B}$ is described in [Press *et al.*, 1992]. In theory the individual values in $\mathbf{B}$ can be chosen arbitrarily. Some choices can clearly be identified as being wrong because the predicted $\mathbf{y}$ is nothing like the data it is supposed to resemble. Other choices seem more probable, but which one is the most likely? The question is: Given a particular set of parameters, what is the probability that the data set obtained could have occurred? The answer

for this question will always be 0 if we do not allow for a small fixed deviation for each y_{ij}, so this will be assumed henceforth.

When an expression for the above probability is determined, given a parameter $\mathbf{B}$, we just maximise that expression. This method gives a *maximum likelihood estimator*.

When It is assumed that the noise matrix $\mathbf{E}$ is normally distributed $N(\mathbf{0}, \sigma^2\mathbf{I})$, see equation (B.2.3). Furthermore, $\mathbf{X}$ is assumed to have full column rank. Then the maximum likelihood estimator of $\mathbf{B}$ is given by [Mardia *et al.*, 1995]

$$\mathbf{B} = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{Y},$$

which is the same solution as found using the least squares method.

One of the qualities of the maximum likelihood estimator is that the expected value of $\hat{\mathbf{b}}$ is equal to the true $\mathbf{b}$. Thus, the estimator is unbiased. The variance is proportional to $(\mathbf{X}^\top\mathbf{X})^{-1}$, which causes problems when $\mathbf{X}^\top\mathbf{X}$ is ill-conditioned. In short we have

$$E(\hat{\mathbf{b}}) = \mathbf{b} \tag{3.52}$$

$$V(\hat{\mathbf{b}}) = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}. \tag{3.53}$$

In the last equation it is assumed that $\mathbf{X}$ is *autoscaled*.

3.4.4.1 Collinearity

Special care must be taken when it is not possible to calculate $(\mathbf{X}^\top\mathbf{X})^{-1}$ as noted above. Also the case where $\mathbf{X}^\top\mathbf{X}$ is *near* singularity can cause numerical problems, because the relative error of the estimated $\mathbf{b}$ depends on the squared condition number of $\mathbf{X}$ [Golub and van Loan, 1991]

$$\frac{\|\hat{\mathbf{b}} - \mathbf{b}\|_2}{\|\mathbf{b}\|_2} \approx u\kappa_2(\mathbf{X}^\top\mathbf{X}) = u\kappa_2(\mathbf{X})^2, \tag{3.54}$$

where u is the unit roundoff ($= 2^{-53} \doteq 1.11 \cdot 10^{-16}$ using double precision) and $kappa_2(\mathbf{X})$ is the condition number of $\mathbf{X}$.

The condition number becomes large if the columns of $\mathbf{X}$ are approximately or exactly linearly dependent and $\mathbf{X}$ is *collinear*. This is often the case when process data is contained in $\mathbf{X}$ because it is based on measurements that depend on a few underlying events only.

One way to avoid collinearity is to select a set of $\mathbf{x}$-variables that do not have the problem of linear dependence. This means that some information/measurements must be disregarded. One method used is *forward selection* where the model starts with one variable and incorporates new ones if they help in explaining $\mathbf{y}$. Another method is called *backward elimination*. Here the model is full in the beginning and variables are eliminated if they do not describe $\mathbf{y}$ well. More advanced methods exist where selected variables can be eliminated at a later stage and vice versa, see [Martens and Næs, 1989] for further information.

If it is possible to eliminate variables in $\mathbf{x}$ without loosing (very much) predictive power, we say that the original data set is *redundant*.

If the data set is highly collinear and not redundant, more advanced methods must be used. These will be described in the following sections.

3.4.5 Ridge Regression

One of the implications of collinearity is that the variance of $\mathbf{b}$ becomes large. That means that although the expected value of our estimator is equal to the true value, we are not sure that we are close to the true value.

In the following $\mathbf{X}$ is assumed to be autoscaled

The distance from $\hat{\mathbf{b}}$ to $\mathbf{b}$ is in the article by Hoerl and Kennard [1970] defined by $L = \|\hat{\mathbf{b}} - \mathbf{b}\|_2$. The expectation is

$$\begin{aligned} E(L^2) &= \sigma^2 \operatorname{tr}(\mathbf{X}^\top \mathbf{X})^{-1} \\ &= \sigma^2 \left(\frac{1}{l_1} + \cdots \frac{1}{l_J} \right) > \frac{\sigma^2}{l_J}, \end{aligned} \tag{3.55}$$

where $l_1, \ldots, l_J$ are the eigenvalues of $\mathbf{X}^\top \mathbf{X}$. It can now be seen that we actually *expect* our estimator to be far away from the true value in the case where we have collinearity, because in that case the last eigenvalue l_J will be small.

Ridge regression (RR) solves the problem of collinearity by solving a nearby problem with the estimator

$$\hat{\mathbf{b}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}, \tag{3.56}$$

where $\lambda \geq 0$. This estimator is biased, but has smaller variance than the maximum likelihood estimator.

There are no general results for choosing λ and it is determined experimentally. E.g. by making a plot of the parameters for different choices of λ. For small λ the parameters will usually vary much. Above a certain value of λ the parameters will be almost constant. The λ chosen of the value that separates these two areas.

According to Frank and Friedman [1993] and the response by Wold [1993] RR is preferred by statisticians because it gives better results when $\mathbf{X}$ is only slightly collinear. When large, very collinear data sets are to be analysed and both $\mathbf{X}$ and $\mathbf{Y}$ contain noise chemometricians prefer other methods that takes these conditions into account. These methods are principal component regression (PCR), see the following section, and projection to latent structures regression (PLS), see section 3.5.

3.4.6 Principal Component Regression

Collinearity is the result of the variables not being uncorrelated. *Principal component analysis* (PCA) is a method, that by linear transformations of the variables obtains uncorrelated variables which are used for the regression [Jackson, 1991].

3.4.7 Regression

PCA is used in regression by once again studying the linear model $\mathbf{Y} = \mathbf{XB} + \mathbf{E}$, but only use the first A columns of $\mathbf{Z}$. This is done without change in notation. The expression for $\mathbf{Z}$ from equation (3.8) is inserted in the linear relationship.

$$\begin{aligned} \mathbf{Y} &= \mathbf{XB} \\ &= \mathbf{ZU}^\top\mathbf{B} \\ &= \mathbf{ZC}, \end{aligned} \tag{3.57}$$

where $\mathbf{C} = \mathbf{U}^\top\mathbf{B}$ is determined by the OLS method:

$$\mathbf{C} = (\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top\mathbf{Y}. \tag{3.58}$$

The original $\mathbf{B}$ parameter matrix can be expressed in terms of the scores

$$\mathbf{B} = \mathbf{U}(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top\mathbf{Y} \tag{3.59}$$

or in terms of the original data set

$$\mathbf{B} = \mathbf{U}(\mathbf{U}^\top\mathbf{X}^\top\mathbf{XU})^{-1}\mathbf{U}^\top\mathbf{X}^\top\mathbf{Y} \tag{3.60}$$

The normal equations (3.58) do not constitute a problem when there are only principal components included in the model corresponding to significantly large eigenvalues.

When all the principal components are used the solution is equal to the OLS solution. When not all components are used the estimator is biased [Jackson, 1991].

3.4.7.1 Selection of Principal Components—Revisited

A rule of thumb has been given that the principal components should be selected according to the size of the accompanying eigenvalue. This is only true when we want to describe variation in $\mathbf{X}$. There is no guarantee that the largest principal components will be the best to describe variation in $\mathbf{Y}$. Examples are given by Jackson [1991] where some of the larger pc's should be eliminated and the smaller ones should be employed in the model because they have larger predictive power.

3.5 Projection to Latent Structures

Projection to latent structures or *partial least squares regression* (PLS) are two names for same regression method. It is used much in chemometrics and especially in the Scandinavian countries, where PLS originates [Wold *et al.*, 2001; Wold, 1993; Martens and Næs, 1989; Geladi and Kowalski, 1986]. The acronym PLS will be used in the following.

The regression method PLS share many of its qualities with PCR. The algorithm works by extracting components that are "optimal" for describing variations in the data. This extraction generates a set of scores and loadings and basically the same possibilities for displaying scores and loadings exist for PLS as for PCA.

In PLS the linear model is

$$\mathbf{X} = \mathbf{T}\mathbf{W}^\top + \mathbf{E} \tag{3.61}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^\top + \mathbf{F}. \tag{3.62}$$

The $\mathbf{W}$ is orthonormal. $\mathbf{T}$ is orthogonal. The relationship between $\mathbf{T}$ and $\mathbf{U}$ is defined by $\mathbf{u}_a = b_a \mathbf{t}_a$ such that the covariance between $\mathbf{u}_a$ and $\mathbf{t}_a$ is maximised. Scores based on $\mathbf{X}$ and $\mathbf{Y}$ are called t-scores and u-scores, respectively. The PLS is different from the ordinary regression methods (OLS, RR, PCR) in the sense that $\mathbf{Y}$ is more directly involved in the determination of the parameter matrix. PLS maximises the covariance between the scores. This ensures that the t-scores in PLS will have larger predictive power than the z-scores found using PCR.

The first scores are found by solving the following problem

$$c = \max(\mathbf{t}^\top \mathbf{u})^2 = \max_{\|\mathbf{w}\|=\|\mathbf{q}\|=1} (\mathbf{w}^\top \mathbf{X}^\top \mathbf{Y}\mathbf{q})^2. \tag{3.63}$$

Compare this with equation (3.23). The solution can be calculated using the SVD. The maximum is equal to the first singular value of $\mathbf{X}^\top\mathbf{Y}$.

The PLS can be calculated by many algorithms. Two algorithms that are used much are the PLS1 and the PLS2 algorithms. PLS1 is used when there is only one variable in $\mathbf{Y}$. PLS2 is used when there are more than one.

The choice of the method depends on the data and what it is going to be used for. If there is only one column in $\mathbf{Y}$ the choice is clear. If there are several a selection has be made. Either the PLS2 algorithm can be used on $\mathbf{Y}$ or the PLS1 algorithm can be used on the individual M columns of $\mathbf{Y}$. If $\mathbf{Y}$ is collinear the latter choice is recommended.

The PLS1 and PLS2 algorithms are based on the NIPALS algorithm that was described in section 3.3.2. The algorithms are described in the following sections.

3.5.1 PLS1: One y-variable

When there is only one column in the response data set the PLS1 algorithm can be used to solve the regression problem. Although the PLS2 algorithm can be used for this regression problem the PLS1 algorithm is much simpler and requires no iterations, but simply extracts components one at a time.

In table 3.2 the PLS1 algorithm from Martens and Næs [1989] is shown. The description of the PLS2 algorithm in the next section also describes the numerical properties of PLS1.

Step	Action	Description
1	$a = 1$	Start with component no. 1
2	$\mathbf{w}_a = \mathbf{X}^\top \mathbf{y} / \mathbf{y}^\top \mathbf{y}$	Regress columns of $\mathbf{X}$ on $\mathbf{y}$
3	$\mathbf{w}_a = \mathbf{w}_a / \|\mathbf{w}_a\|$	Normalise $\mathbf{w}$
4	$\mathbf{t}_a = \mathbf{X}\mathbf{w}_a$	Calculate scores
5	$\mathbf{p}_a = \mathbf{X}^\top \mathbf{t}_a / \mathbf{t}_a^\top \mathbf{t}_a$	Calculate loadings
6	$q_a = \mathbf{y}^\top \mathbf{t}_a / \mathbf{t}_a^\top \mathbf{t}_a$	Regress $\mathbf{y}$ on $\mathbf{t}$
7	$\mathbf{X}_{new} = \mathbf{X}_{old} - \mathbf{t}_a \mathbf{p}_a^\top$	Calculate residuals
8	$\mathbf{y}_{new} = \mathbf{y}_{old} - \mathbf{t}_a q_a$	Calculate residuals
9	$a = a + 1$, $a \leq A$? goto 2	Select one more component

Table 3.2. PLS1 algorithm for one $\mathbf{y}$-variable. $\mathbf{X}$ and $\mathbf{y}$ should be centred. A is the total number of components.

3.5.2 PLS2: More than one y-variable

The algorithm for the case where we have more than just one $\mathbf{y}$-variable is more commonly seen [Martens and Næs, 1989; MacGregor *et al.*, 1994; Höskuldsson, 1988]. It is called the PLS2 algorithm and is shown in table 3.3. It has to use iterations in order to give maximal vectors.

We now assume the inner loop (3–7) has obtained convergence of $\mathbf{u}$. The length of $\mathbf{w}$ in step 3 is called c. The defining statement of $\mathbf{u}$ in step 7 is

Step	Procedure	Description
1	$a = 1$	Start with the first component
2	$\mathbf{u}_a = \mathbf{y}_{any}$	Set $\mathbf{u}$ = any column of $\mathbf{Y}$
3	$\mathbf{w}_a = \mathbf{X}^\top \mathbf{u}_a / \mathbf{u}_a^\top \mathbf{u}_a$	Regress columns of $\mathbf{X}$ on $\mathbf{u}$
4	$\mathbf{w}_a = \mathbf{w}_a / \|\mathbf{w}_a\|$	Normalise $\mathbf{w}$
5	$\mathbf{t}_a = \mathbf{X}\mathbf{w}_a$	Calculate scores
6	$\mathbf{q}_a = \mathbf{Y}^\top \mathbf{t}_a / \mathbf{t}_a^\top \mathbf{t}_a$	Regress $\mathbf{Y}$ on $\mathbf{t}$
7	$\mathbf{u}_a = \mathbf{Y}\mathbf{q}_a / \mathbf{q}_a^\top \mathbf{q}_a$	Calculate scores for $\mathbf{Y}$
8		If $\mathbf{u}$ not conv. goto 3
9	$\mathbf{p}_a = \mathbf{X}^\top \mathbf{t}_a / \mathbf{t}_a^\top \mathbf{t}_a$	Calculate $\mathbf{X}$ loadings
10	$\mathbf{X}_{new} = \mathbf{X}_{old} - \mathbf{t}_a \mathbf{p}_a^\top$	Calculate residuals
11	$\mathbf{Y}_{new} = \mathbf{Y}_{old} - \mathbf{t}_a \mathbf{q}_a^\top$	Calculate residuals
12	$a = a + 1$, $a \leq A$? goto 3	Select one more component

Table 3.3. PLS2 algorithm for more than one $\mathbf{y}$-variable. $\mathbf{X}$ and $\mathbf{Y}$ should be centred. A is the total number of components.

substituted (backwards) with the preceding defining statements:

$$\begin{aligned}\mathbf{u} &= \mathbf{Yq}/[\mathbf{q}^\top\mathbf{q}] \\ &= \mathbf{YY}^\top\mathbf{t}/[(\mathbf{q}^\top\mathbf{q})(\mathbf{t}^\top\mathbf{t})] \\ &= \mathbf{YY}^\top\mathbf{Xw}/[(\mathbf{q}^\top\mathbf{q})(\mathbf{t}^\top\mathbf{t})] \\ &= \mathbf{YY}^\top\mathbf{XX}^\top\mathbf{u}/[(\mathbf{q}^\top\mathbf{q})(\mathbf{t}^\top\mathbf{t})(\mathbf{u}^\top\mathbf{u})c].\end{aligned} \tag{3.64}$$

The expression defines an eigenproblem, see section A.3. A value a is defined as the eigenvalue. It is the denominator of (3.64). Likewise, it is possible to define eigenproblems in terms of $\mathbf{q}$, $\mathbf{t}$ and $\mathbf{w}$ [Höskuldsson, 1988]

$$\mathbf{YY}^\top\mathbf{XX}^\top\mathbf{u} = a\mathbf{u} \tag{3.65}$$

$$\mathbf{Y}^\top\mathbf{XX}^\top\mathbf{Yq} = a\mathbf{q} \tag{3.66}$$

$$\mathbf{XX}^\top\mathbf{YY}^\top\mathbf{t} = a\mathbf{t} \tag{3.67}$$

$$\mathbf{X}^\top\mathbf{YY}^\top\mathbf{Xw} = a\mathbf{w}. \tag{3.68}$$

The eigenvalues (a) are equal to the singular values of the matrix $\mathbf{X}^\top\mathbf{Y}$.

The vectors in the algorithm and in the eigenproblems are illustrated in figure 3.7.

3.5.3 Regression

The parameter matrix $\mathbf{B}$ in the linear model $\mathbf{Y} = \mathbf{XB} + \mathbf{F}$ can be defined using the matrices in the PLS2 algorithm as [Martens and Næs, 1989]

$$\mathbf{B} = \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}\mathbf{Q}^\top \tag{3.69}$$

It is believed that this estimate of the parameters is more robust than the estimates obtained from OLS, RR and to some degree PCR when the data matrices are large and collinear and there is noise in both $\mathbf{X}$ and $\mathbf{Y}$. The reason being that the assumptions in PLS are more relaxed [Wold, 1993] as outlined in table 3.4.

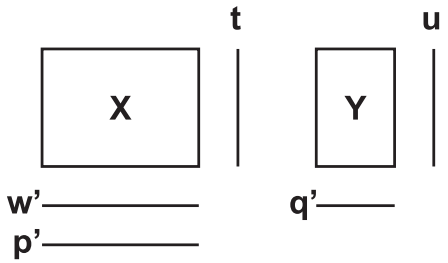
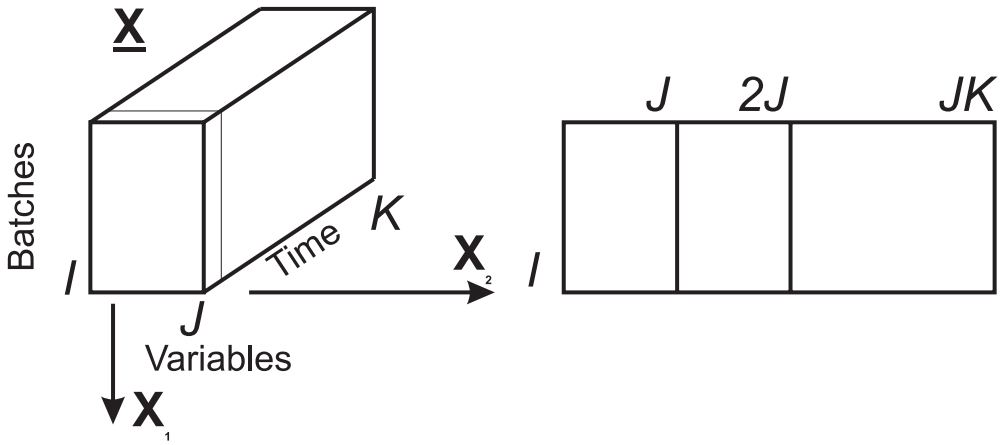


Figure 3.7. Vectors and matrices in PLS iteration

OLS, RR	PCA, PLS	Remarks
x_j independent	x_j correlated and lumped (not independent)	Latent variables
x_k exact	x_k may have errors	**X** incomplete and noisy
Model "true" (residuals random)	Residuals may have structure	**X** may have structure unrelated to **Y**
Data homogeneous	Data homogeneous	Relation $\mathbf{X} \rightarrow \mathbf{Y}$ same throughout investigated region of **X** space

Table 3.4. Assumptions underlying the linear modelling methods [Wold, 1993]. The third column describes the improvement that the PCA and PLS methods have over the traditional regression methods.

3.6 Unfolding



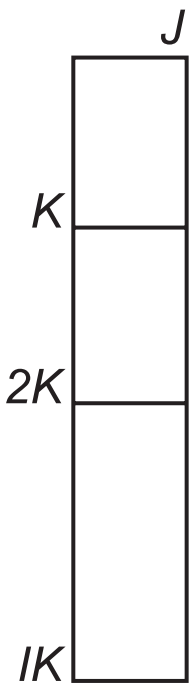


Figure 3.8. The principle of unfolding a three-way matrix.

When data from a batch or fed-batch is to be analysed it can conveniently be put in a *three-way matrix* $\underline{\mathbf{X}}$. The requirements are that the duration of the batches has to be equal. Since batch operations usually or at least ideally are performed in the same way every time, the average batches will be of approximately the same length and any minor difference can be eliminated by truncation. Other ways have been studied that handle the problem of varying batch durations in greater detail. One such method is dynamic time warping (DTW) [Gollmer and Posten, 1996]. DTW is computer intensive and is not widely used. Another method of scaling the batch length is to find a monotonely increasing variable and use that as time indicator. This method has been used with some success [Westerhuis *et al.*, 1999].

The number of measurements is J, the number of time samples is K and the number of batches is I.

In order to do normal matrix manipulations with the data, the matrix $\underline{\mathbf{X}}$ is turned into a two-way matrix. This is called *unfolding*. This matrix is cut into slices. The slices can be cut in three different ways. Slices can be cut parallel to the JK-, IK- or IJ-axes. Slices can be assembled either vertically or horizontally. This leads to a total of six configurations and one is chosen depending on the relations to be investigated. For process chemometrics applications the slices are cut in the matrix as shown in figure 3.8. The slices can be put together in two different ways resulting in either of the two matrices $\mathbf{X}_1$ or $\mathbf{X}_2$.

When the data set has been unfolded the PCA can be

made on either $\mathbf{X}_1$ or $\mathbf{X}_2$. The two configurations lead to different kinds of analyses.

The $\mathbf{X}_1$ matrix corresponds to the usual way of using PCA where the columns contain the original measurement variables. The covariance matrix considered in this case describes the covariance between the original variables and is thus small and readily interpretable. Likewise the eigenvectors will be in terms of the original variables.

When $\mathbf{X}_2$ is considered we study the difference between batches. A row is filled with all the data from a batch. Each column contains a certain measurement variable at a certain time, but from different batches. If this configuration is going to make any sense, the batches has to be run in almost the same way. If this assumption is fulfilled the mean of each column can be used as a normal value and the variation of the data around this point provide on single variable confidence intervals for normal behaviour of the process.

In order to distinguish the two methods the method where $\mathbf{X}_2$ is considered will be prefixed with "multi-way", e.g. multi-way principal component analysis (MPCA) and multi-way partial least squares (MPLS).

3.6.1 Three-Way Principal Component Analysis

Principal component analysis can be generalised to matrices with more than two dimensions. This can be done by including factors such as different batches, fermentors or microorganism strains as new dimensions to the original data structure (variables versus time).

The number of dimensions of the data matrix is in data analysis referred to as *ways* or *modes*. This is done to avoid confusion with the term *dimension* used to describe vector spaces. The term *ways* will be used here.

3.6.2 Multi-Way PCA

When the number of ways rises, the calculations become more complex. Three-way PCA has been investigated by Kroonenberg [1983].

PARAFAC and Tucker models are some of the true multi-way modelling methods [Louwerse and Smilde, 2000; Westerhuis *et al.*, 1999]. These decompositions are shown in figure 3.9.

If the PCA decomposition is written as

$$x_{im} = \sum_{r=1}^{R} a_{ir} b_{mr} + e_{im} \tag{3.70}$$

the corresponding PARAFAC decomposition may be written as

$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} + e_{ijk} \tag{3.71}$$

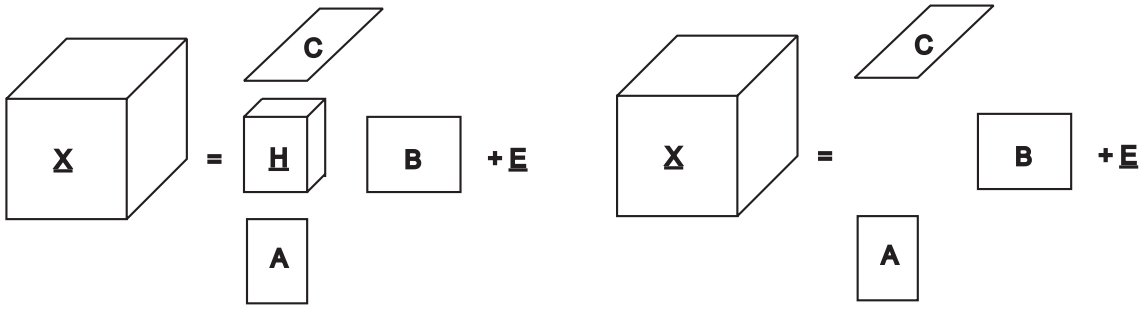


Figure 3.9. Tucker3 (left) and PARAFAC (right) decompositions

and the Tucker decomposition may be written as

$$x_{ijk} = \sum_{r=1}^{R} \sum_{s=1}^{S} \sum_{t=1}^{T} a_{ir} a_{js} c_{kt} h_{rst} + e_{ijk} \tag{3.72}$$

These models that directly utilise the three-way structure of the batch data have fewer parameters than methods using unfolding. This is often seen as an advantage to true three-way modelling techniques, but since the parameters in none of the modelling techniques are indenpendent it may not be so important.

PCA of the unfolded matrix $\mathbf{X}$ is equivalent to MPCA on $\underline{\mathbf{X}}$ [Wold *et al.*, 1987a; Nomikos, 1995]. Therefore in most succesful applications the unfolding technique is used and this method is also the only described and use din this thesis.

Recent applications of true multi-way methods for fault diagnosis may be found in [Boqué and Smilde, 1999; Louwerse and Smilde, 2000; Bro, 1997].

Working with matrices with more than three ways (MPCA) is possible, but is difficult [Jackson, 1991; Wold *et al.*, 1987a]. "Users are advised to proceed beyond three-way data with considerable caution" [Coxon, 1982].

3.7 Fault Diagnosis

Control and monitoring of continuous processes are frequently based on the idea that the variables should be constant, i.e. they should follow a setpoint which is kept constant most of time.

It is the purpose of statistical process control to monitor the process and detect any deviation from the setpoints that is significant.

An introduction to the statistical process control of these kind of processes can be found in [Pond, 1994]. An overview of multivariate statistical process control is provided by [Çinar and Undey, 1999; MacGregor and Kourti, 1995].

Fault diagnosis involves three tasks:

- Detection

- Isolation
- Identification

Fault detection is the task of detection when a fault occurs in the plant. Faults must be detected early and reliably in order for the fault diagnosis system to be useful. The fault can only be handled after detection of the fault . The first step after a fault has been detected is to isolate the fault. For large plants this task involves finding the place in the plant where the fault has its origin and ultimately which variables that indicate the faulty behaviour. Fault identification is a task that involves process knowledge in order to find the physical reason for the fault and finding a strategy to compensate or eliminate the fault, if possible. The detection and isolation tasks can be automated by the methods described in the following sections. As the fault identification tasks require process knowledge of a faulty plant it is much more complicated to automate and in this thesis this task is left to the process operator or systems engineer.

3.7.1 PCA

The analysis of the fed-batch process using PCA demands that the data set $\underline{\mathbf{X}}$ is unfolded into $\mathbf{X}_1$ $(IK \times J)$ as shown in figure 3.8. This unfolded matrix will in the following be called $\mathbf{X}$ and it will be assumed autoscaled.

From the measurements obtained at a certain point in time $\mathbf{x}$ $(J \times 1)$ the scores can be calculated

$$\mathbf{z} = \mathbf{U}^\top \mathbf{x}. \tag{3.73}$$

Using the PCA model it is possible to transform the scores back into the original variables

$$\mathbf{x} = \mathbf{U}\mathbf{z}. \tag{3.74}$$

There is only exact correspondence between the variables and the scores when all the pc's are used. When the number of pc's is reduced some error is introduced, but if the model has been correctly made the residual will be small and only include unwanted noise.
The predicted $\mathbf{x}$ when A pc's are used is given by

$$\hat{\mathbf{x}} = \tilde{\mathbf{U}}\tilde{\mathbf{z}}, \tag{3.75}$$

where $\tilde{\mathbf{t}} = \tilde{\mathbf{W}}^\top \mathbf{x}$ and the tildes are used to denote reduced matrices and scores. By introduction of $\mathbf{x}$ the equation can be rewritten into

$$\mathbf{x} = \tilde{\mathbf{U}}\tilde{\mathbf{t}} + (\mathbf{x} - \hat{\mathbf{x}}). \tag{3.76}$$

The first term on the right hand side is the contribution of the model. The second is a residual: The difference between the model and the correct value.

3.7.2 Q-Statistic

If the PCA model uses all components it is possible to invert the model using equations (3.2) and (3.12) to obtain

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{U}\mathbf{z} \tag{3.77}$$

and

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{V}\mathbf{y} \tag{3.78}$$

It is thus possible to get the original variables from the z- and y-scores. This can only be done when all the principal components are used.

If we define $\tilde{\mathbf{U}}$ as the matrix consisting of the first A columns of $\mathbf{U}$ we can calculate an estimate of $\mathbf{x}$ called $\hat{\mathbf{x}}$:

$$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \tilde{\mathbf{U}}\mathbf{z}. \tag{3.79}$$

This can be rewritten to

$$\mathbf{x} = \bar{\mathbf{x}} + \tilde{\mathbf{U}}\mathbf{z} + (\mathbf{x} - \hat{\mathbf{x}}), \tag{3.80}$$

which is a linear model. The first term is the contribution of the mean, the second term represents the contribution from the pc's and the third term represents anything not explained by the pc-model—the residual. The Q-statistic is used to monitor the residual.

The Q-statistic is defined as the square sum of the residuals:

$$\begin{aligned} Q &= (\mathbf{x} - \hat{\mathbf{x}})^\top(\mathbf{x} - \hat{\mathbf{x}}) && (3.81)\\ &= \mathbf{x}^\top(\mathbf{I} - \tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top)\mathbf{x} \\ &= \mathbf{z}^\top\mathbf{U}^\top\mathbf{U}\mathbf{z} - \tilde{\mathbf{z}}^\top\tilde{\mathbf{z}} \\ &= \sum_{j=1}^{J} z_j^2 - \sum_{j=1}^{A} z_j^2 \\ &= \sum_{j=A+1}^{J} z_j^2, && (3.82) \end{aligned}$$

where z_i is calculated using $\tilde{\mathbf{U}}$ obtained from autoscaled data. Thus the z-scores are given by $\mathbf{z} = \tilde{\mathbf{U}}^\top\mathbf{D}^{-1}(\mathbf{x} - \bar{\mathbf{x}})$, where $\mathbf{D}$ is defined in equation B.15 on page 180.

To obtain an upper limit for Q a test may be performed [Jackson, 1991]:

$$\Theta_1 = \sum_{i=A+1}^{J} l_i \tag{3.83}$$

$$\Theta_2 = \sum_{i=A+1}^{J} l_i^2 \tag{3.84}$$

$$\Theta_3 = \sum_{i=A+1}^{J} l_i^3 \tag{3.85}$$

$$h_0 = 1 - \frac{2\Theta_1\Theta_3}{3\Theta_2^2} \tag{3.86}$$

The quantity

$$c = \Theta_1 \frac{\left(\frac{Q}{\Theta_1}\right)^{h_0} - \frac{\Theta_2 h_0(h_0-1)}{\Theta_1^2} - 1}{\sqrt{2\Theta_2 h_0^2}} \tag{3.87}$$

is approximately normally distributed with zero mean and unit variance. The critical value for Q becomes:

$$Q_\alpha = \Theta_1 \left(\frac{c_\alpha \sqrt{2\Theta_2 h_0^2}}{\Theta_1} + \frac{\Theta_2 h_0(h_0-1)}{\Theta_1^2} + 1 \right)^{1/h_0}, \tag{3.88}$$

where c_α is the normal deviate cutting off an area under the upper tail of the distribution if h_0 is positive and under the lower if h_0 is negative. This distribution holds whether or not all of the significant components are used and even if some nonsignificant components are employed.

The test described can only be used if all eigenvalues have been obtained. If that is not the case the following expressions for the Θ's can be used instead

$$\mathbf{S}_E = \mathbf{E}\mathbf{E}^\top/(I-1) \tag{3.89}$$

$$\begin{aligned} \Theta_1 &= \mathrm{tr}(\mathbf{S}_E) \\ \Theta_2 &= \mathrm{tr}(\mathbf{S}_E^2) \\ \Theta_3 &= \mathrm{tr}(\mathbf{S}_E^3), \end{aligned} \tag{3.90}$$

where $\mathbf{E}$ is defined by equation (3.6) as $\mathbf{E} = \mathbf{X} - \mathbf{Z}\mathbf{U}$. The remaining part of the test stays the same.

3.7.2.1 Detection

The Q-statistic has been used by Saner and Stephanopoulos [1992] in fault detection. The PCA model is based on data sets obtained from batches which can be used as a reference for the fault detection. This data set is called the normal data set. Based on this data set the confidence limits are calculated.

In an on-line operation the Q-statistic is calculated each time a new measurement has been obtained. If the Q-statistic is above the acceptable limit a fault is detected.

3.7.2.2 Isolation

When a fault has been detected it is important that the cause of the fault can be found. A step in this direction is isolation, where the deviating variables are identified.
The isolation is based on equation (3.81), which is expressed as a sum.

$$Q = (\mathbf{x} - \hat{\mathbf{x}})^\top (\mathbf{x} - \hat{\mathbf{x}}) = \sum_{j=1}^{J} (x_j - \hat{x}_j)^2. \tag{3.91}$$

It can be seen that the contribution of each variable is a term in the sum. A plot of the contributions can be made. The deviating variables will easily be spotted in such a plot and the fault will thus be isolated.

3.7.3 MPCA

In this thesis multi-way principal component analysis (MPCA) will be used to denote a conventional PCA on a two-way matrix that has been obtained from an unfolding of a three-way matrix as shown for the matrix $\mathbf{X}_2$ $(I \times JK)$ in figure 3.8.

The detection of errors using MPCA is based on the assumption that the normal data set is obtained from batches that are run similarly and where the variables are expected to have a certain value at a certain point in time. The MPCA model is used to quantify the deviation from this trajectory.

The model can be validated using the Hotelling T^2 statistic and the aforementioned Q-statistic.

The calculations and interpretations can in principle be carried out in exactly the same way as conventional PCA, but the dimension op the data matrix $\mathbf{X}$ is different (wide and short) and special care must be taken to keep matrices small during computations. Commercial chemometrics software not made for this type of data is especially cumbersome to work with because they do not support this matrix structure.

3.7.4 Interpretation of the Principal Components

In order to interpret the PCA model the relationship between individual columns in $\mathbf{X}$ and the y-scores is examined. The fraction of variation of $\mathbf{x}_j$ explained by $\mathbf{y}_a$ is v_{ja}^2 [Jackson, 1991]. This relationship will be explored in the following.

The total variance explained by a y-score $\mathbf{y}_a$ is given by the sum

$$l_a = \sum_{j=1}^{JK} v_{ja}^2 = \mathbf{v}_a^\top \mathbf{v}_a. \tag{3.92}$$

and is the well known relation $\mathbf{V}^\top\mathbf{V} = \mathbf{L}$ (equation (3.13)). The variance of the variables explained by the ath pc is given by $\mathbf{v}_a\mathbf{v}_a^\top$. The variance unexplained by the first score is $\mathbf{S} - \mathbf{v}_1\mathbf{v}_1^\top$. The results for the subsequent scores are found by summation until the relation $\mathbf{V}\mathbf{V}^\top = \mathbf{S}$ is found.

3.7.4.1 Explanation of Variance by each Principal Component as a Function of Time

Results will be given for the first principal component only. The contribution of the other pc's are calculated similarly.

The first column of $\mathbf{V}$ corresponding to the first pc can be decomposed into K groups of J elements corresponding to the variables at a given point in time.

$$\mathbf{v}_1 = \begin{bmatrix} \vdots \\ v_{(k-1)J+1,1} \\ v_{(k-1)J+2,1} \\ \vdots \\ v_{(k-1)J+J,1} \\ \vdots \end{bmatrix}$$

The fraction of total explained variance for the first pc at the time point k is then given by

$$\frac{1}{J}\sum_{j=1}^{J} v_{(k-1)J+j,1}^2 \tag{3.93}$$

The result can be generalised to all pc's by using the relevant columns of $\mathbf{V}$.

3.7.4.2 Contribution by the Measurement Variables to the Principal Components

It is also possible to decompose $\mathbf{v}_1$ in the following way where the elements corresponding to a certain variable at all points in time are extracted

$$\mathbf{v}_1 = \begin{bmatrix} \vdots \\ v_{j,1} \\ \vdots \\ v_{J+j,1} \\ \vdots \\ \vdots \\ v_{(K-1)J+j,1} \\ \vdots \end{bmatrix}$$

The fraction explained by the first pc of the variation in the variable j is then

$$\frac{1}{K}\sum_{k=1}^{K} v^2_{(k-1)J+j,1}. \tag{3.94}$$

3.7.5 T^2 statistic

The T^2 statistic is an overall measure of the distance between the mean of normal data set and the individual measurements.
The y-score for the ith batch is obtained by

$$\mathbf{y} = \mathbf{W}^\top \mathbf{X}_{(i)}, \tag{3.95}$$

where $\mathbf{W}$ is defined in (3.10) by $\mathbf{W} = \mathbf{U}\mathbf{L}^{-1/2}$.
The T^2 statistic is defined by

$$T^2 = \mathbf{y}^\top \mathbf{y}. \tag{3.96}$$

When the data set that is going to be tested is included in the one used to produce $\mathbf{W}$ the Hotelling T^2 statistic has a beta distribution (times a factor) [Tracy and Young, 1992]

$$T^2 \sim \frac{(I-1)^2}{I} B_{A/2,(I-A-1)/2} \tag{3.97}$$

The relation above holds only when $\mathbf{y}$ is normally distributed, which is approximately satisfied due to the central limit theorem.

Plots of the y-scores from the ith and jth principal component for all batches in the normal data set can be made. The confidence circle[2] for the scores has radius

$$\left(\frac{(I-1)^2}{I} B_{A/2,(I-A-1)/2,\alpha}\right)^{1/2}. \tag{3.98}$$

In order to acknowledge the data set as normal, the scores should stay inside the circle at a high significance level and the scores should be evenly distributed. Any "holes" or uncovered areas in the circles suggest that the data set is not representative or that the MPCA-model is not able to explain the variation of the process. For real process data and a low number of batches in the normal data set it is often the case that patterns not resembling a circle occur. A special confidence region can then be constructed using kernel density estimation (see section 3.7.7. During model development it is recommended, though, that confidence circles are used.

[2]Multivariate analysis of normally distributed variables often lead to confidence ellipses, but since the y-scores have been scaled to unit variance the confidence areas become circular.

3.7.6 On-line Estimation of t-scores

When a data set for a new batch have been obtained the t-scores can be calculated

$$\mathbf{t} = \mathbf{W}^{\top}\mathbf{X}_{new}^{\top}, \tag{3.99}$$

where $\mathbf{X}_{new}$ is the $(1 \times JK)$ matrix that contains the new data set.
A problem arises when this method is to be used on-line on a running process simply because $\mathbf{X}_{new}$ does not have the proper size until the process has finished.

The most correct way to overcome the problem of $\mathbf{X}_{new}$ not having the right size is to calculate a $\mathbf{U}$ matrix corresponding to every point in time of the batch process. This will require a significant amount of storage and therefore the method is not used. We can also use the data already obtained from the batch to "fill in the blanks" in the remaining part of the data matrix. Three methods of "filling in the blanks" of $\mathbf{X}_{new}$ have been described in [Nomikos and MacGregor, 1995]. The three methods are:

- Use the PCA or PLS model to estimate the unknown data.
- Fill in the blanks with zeros.
- Fill in the blanks by repeating the last obtained measurements to the remaining part of the data matrix.

There is little difference between the methods in terms of accuracy and robustness, especially towards the end of the batch. The last method gives faster response in the event of a fault, which is a nice feature. This method is expected to give a poor estimation result in the beginning of a batch because of lack of information, but has been found to give a good result when more than $1/3$ of the batch history have been obtained [Gregersen and Jørgensen, 1999].

3.7.7 Kernel density estimate of confidence area

Confidence limits for the y-scores can be established using the circles defined by (3.98). These limits are suitable when developing the model since the data used in the development are all from similar batches (in some sense). When the model has been based on only a few data sets it is often seen that the circles are not filled evenly with data points. This can be due to violation of the normality assumptions of the scores or simply a result of basing the model on historic data that do not fill the entire score space because the data were obtained under normal operation conditions as opposed to under conditions where the operation has been planned to fill the operating window evenly. It has been proposed that kernel density estimates of the confidence limits can be beneficial when monitoring the process [Martin *et al.*, 1996]. Using kernel density estimates for the limits it is assured that confidence is only given to scores that are in an operating region that has been explored when building the model.

An general estimator for the kernel density can be defined as [Simonoff, 1996]

$$f(\mathbf{t}) = \frac{1}{n|\mathbf{H}|} \sum_{i=1}^{I} K_d \left(H^{-1}(\mathbf{t} - \mathbf{t}_i) \right), \tag{3.100}$$

where $|\mathbf{H}|$ is the absolute value of the determinant of the matrix $\mathbf{H}$, which is a bandwidth matrix. K_d is the multivariate kernel function. One way of creating K_d from a univariate kernel K is by using a product kernel

$$K_d(\mathbf{u}) = \prod_{j=1}^{d} K(u_j). \tag{3.101}$$

The Gaussian kernel $K(u) = (2\pi)^{-1/2} \mathrm{e}^{-u^2/2}$ will be used here. Other kernels may be used, but the resulting plots do not vary much when other kernels are chosen. The bandwidth matrix is chosen to be a diagonal matrix. This complies with the scores being independent. The bandwidth has to be chosen based on the shape of the data and the detail that is desired in the plot. A default value for the bandwidth is $h_i = 1.059 s_i I^{-1/5}$, where s_i is the standard deviation of the ith score [Simonoff, 1996]. For confidence limits in the score plots it has been found that the bandwidth usually has to be slightly larger than the default value leading to a less detailed confidence region yielding smooth contours.

3.7.8 Squared Prediction Error

The Q-statistic can be used to evaluate future observations, but has been found to give a poor instantaneous estimate of the state of the process. Instead the *Squared prediction error* (SPE) is used. It is defined by

$$\mathrm{SPE}_k = \sum_{r=(k-1)J+1}^{Jk} \mathbf{e}_{(r)}^2, \tag{3.102}$$

where $\mathbf{e}_{(r)}$ is the rth element of the residual matrix defined by

$$\mathbf{E} = \mathbf{X}_{new} - \mathbf{z}^\top \mathbf{U}^\top. \tag{3.103}$$

The distribution of the SPE can be approximated by a weighted χ^2 distribution

$$\mathrm{SPE}_k \sim \frac{v_k}{2m_k} \chi^2_{2m_k^2/v_k}, \tag{3.104}$$

where m_k and v_k are the mean and variance of the SPE obtained from the normal data set at time instant k [Nomikos, 1995].

3.7.8.1 Detection

The SPE will show if the underlying process is different from the model. Any new behaviour that was not present in the data set used to make the PCA-model will be pointed out by the SPE.

3.7.8.2 Isolation

The contribution by the variables to the SPE can be calculated similarly to the Q-statistic

$$\text{SPE}_k = \sum_{r=J(k-1)+1}^{Jk} \mathbf{e}_r^2 = \mathbf{e}_{((k-1)J+1)} + \cdots + \mathbf{e}_{(Jk)}$$

is just a question of extracting the corresponding term from the sum of the J numbers.

3.7.9 T_f^2 statistic

3.7.9.1 Detection

The Hotelling statistic can be calculated from a future independent y-score in the same way as stated earlier in section 3.7.5. The quantity will be called T_f^2. The distribution is [Tracy and Young, 1992]

$$T_f^2 \sim \frac{A(I^2-1)}{I(I-A)} F_{A,I-A} \tag{3.105}$$

Again the distribution relies on the assumption that the y-scores are normally distributed.

The radius of the confidence circles of the y-score plot can be calculated from

$$\left(\frac{A(I^2-1)}{I(I-A)} F_{A,I-A} \right)^{1/2} \tag{3.106}$$

3.7.9.2 Isolation

Contribution plots for the T_f^2 statistic is readily made when the contribution is to be found in terms of the scores because of the definition of the T_f^2 statistic

$$T_f^2 = \mathbf{y}^\top \mathbf{y} = \sum_{a=1}^{A} y_a^2.$$

The contribution from each score is then just the corresponding term in the sum.

The contribution from the variables require more work. The scores calculated by

$$\mathbf{t}_k = (\mathbf{W}_{1:Jk}^\top \mathbf{W}_{1:Jk})^{-1} \mathbf{W}_{1:Jk}^\top \mathbf{X}_{new,k}^\top = \mathbf{A}_k \mathbf{t}_{f,k}, \tag{3.107}$$

where $\mathbf{A}_k = (\mathbf{W}_{1:Jk}^\top \mathbf{W}_{1:Jk})^{-1}$ and

$$\mathbf{t}_{f,k} = \mathbf{W}_{1:Jk}^\top \mathbf{X}_{new,k}^\top = \sum_{n=1}^{Jk} \mathbf{w}_n \mathbf{x}_{new,(n)}. \tag{3.108}$$

The sum can be split up according to variables and the contribution can be calculated as in Mason *et al.* [1995], where PCA is not employed.
$\mathbf{t}_{f,k}$ can be written as a double sum

$$\mathbf{t}_{f,k} = \sum_{j=1}^{J}\sum_{n=1}^{k} \mathbf{w}_{(n-1)J+j}\mathbf{x}_{new,((n-1)J+j)} = \sum_{j=1}^{J} \mathbf{t}_{f,j} \tag{3.109}$$

where the terms in the inner sum is the contribution to the score $\mathbf{t}_{f,k}$ by variable j. The contribution to the y-scores can be written as $\mathbf{y}_{f,j} = \mathbf{L}^{-1/2}\mathbf{A}\mathbf{t}_{f,j}$.
Contribution to the T_f^2 statistic from the j variable is then

$$T_{f,j}^2 = \mathbf{y}_{f,j}^{\top}\mathbf{y}_{f,j} \tag{3.110}$$

3.8 PLS—Fault Diagnosis

Fault diagnosis using MPLS is based on the same principles as fault diagnosis using MPCA. Many of the results in this section will therefore look familiar.

When a model of a system is made using PCA we hope that the information content in the data is sufficient to account for the natural variation in the data and that the PCA model is able to describe that variation. When it comes to detection of faults the PCA model will signal a fault if the variables are outside there normal operating range. This is based on the idea that the process should be operated in the same way every time and that normal operation is optimal in some sense. Thus, any deviation from the recipe will be less than optimal, i.e. a fault.

When a dependent variable ($\mathbf{Y}$) is available in addition to the independent ($\mathbf{X}$) variable a PLS model may be developed instead of (or in addition to) a PCA model. Thus, for processes where there are no quality measures a PCA model can only be built. This is e.g. relevant for difficult to quantifiable features such as flavour or complex mixture compositions. When one or more quantifiable quality variables can be obtained they will usually be utilised in a PLS model since it then becomes possible to target the fault diagnosis toward faults that affect the quality of the process.

When PLS is used for fault diagnosis it is possible not only to base the analysis on the scores, but also the predicted values can be used in the analysis.

The data used are the on-line measurements ($\underline{\mathbf{X}}$) as independent variables and the dependent variable ($\mathbf{Y}$) contains some quality variable of interest measured at the end of the batch. This can either be some kind of concentration variable or the yield of the product of by-products.

The fault analysis will be carried out as outlined in the article by Kourti *et al.* [1995].

A PLS model is defined by the linear relationships between the independent

variable $\mathbf{X}$ and the dependent variable $\mathbf{Y}$.

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{W}^\top + \mathbf{E} \\ \mathbf{Y} &= \mathbf{U}\mathbf{Q}^\top + \mathbf{F} \end{aligned} \qquad \mathbf{u}_a = b_a \mathbf{t}_a. \tag{3.111}$$

The usual linear regression is defined by

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{F}, \tag{3.112}$$

which is a rewrite of (3.111) that in many cases is easier to work with. The regression parameter $\mathbf{B}$ have been shown to be

$$\mathbf{B} = \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}\mathbf{Q}^\top. \tag{3.113}$$

The matrices $\mathbf{W}$, $\mathbf{P}$ and $\mathbf{Q}$ are determined by the algorithms PLS1 and PLS2 described in section 3.4.

When a PLS model is used for prediction we use

$$\begin{aligned} \mathbf{Y} &= \mathbf{X}\mathbf{B} && (3.114) \\ &= \mathbf{X}\mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}\mathbf{Q}^\top \\ &= \mathbf{T}\mathbf{Q}^\top. && (3.115) \end{aligned}$$

Equation (3.114) can be used for finding the influence of the variables upon the dependent variable, whereas equation (3.115) can be used to find the relationship between scores and the dependent variable.

3.8.1 Fault Detection and Isolation

Detection and isolation of faults is based on monitoring of the scores.

On-line data from a new batch is collected and unfolded into a matrix $\mathbf{X}_{new}$ $(1 \times KJ)$, where K is the number of time samples and J is the number of variables. This matrix must be scaled in the same way as the data used for making the MPLS model.

In an on-line situation we have the same problems as in the MPCA case: The matrix $\mathbf{X}_{new}$ is not full until the batch has finished. The empty part of $\mathbf{X}_{new}$ is constructed by using the information available from the beginning of the fermentation *up until* time k, and filling the remaining part of $\mathbf{X}_{new}$ with the same measurements as the one obtained *at* time k.

Based on the data in $\mathbf{X}_{new}$ the t-scores can be calculated

$$\begin{aligned} \mathbf{t}_{new} &= \left(\mathbf{X}_{new}\mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}\right)^\top \\ &= \mathbf{C}^\top\mathbf{X}_{new}^\top, \end{aligned} \tag{3.116}$$

where $\mathbf{C} = \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}$.

The t-scores are a measure of the variation of the process in a reduced space. If all the elements of the score are zero, i.e. the score is at the origin, the product concentration is predicted to be the mean value of the normal data set. If this mean value is satisfactory then a strategy equal to the one used in MPCA is useful. If the score deviates (too much) from the origin then the process has a fault, which has to be handled.

The MPLS analysis differs from MPCA in the weights that are put on the different measurements. In MPLS only variables that have influence on the dependent variable are allowed to signal a fault. In MPCA it is possible for all variables, even irrelevant ones, to signal a fault.

A score plot will make it possible for the process operator to monitor the process. Based on the t-scores obtained from the normal data set confidence regions can be calculated and compared with the scores.

Based on experience the scores may be interpreted just like in the MPCA case. This would allow the process operator to see not only where the process is but also in what direction it is moving. This predictive capability would allow intervention before a fault has actually developed.

By equation (3.115) it can be seen that the t-scores are closely related to the dependent variable. The values in $\mathbf{Q}$ determine the weight the scores has on the dependent variable. These values can all be chosen to be positive by changing sign the appropriate places in the other matrices defined by the PLS algorithms. Positive values in $\mathbf{Q}$ will make is easy to monitor the dependent variable using the score plot. If a large dependent variable is desirable then large positive scores would be too.

Two criteria (at least) are available for monitoring the process using the t-scores. The first is to keep the scores around the origin assuming the model indeed is based on good batches, where good means in terms of the quality of the process, e.g. high productivity. Using only good batches for model development makes the model rather poor at predicting the dependent variable hence a somewhat broad data set has to be used. When such a model is used then a second strategy can be used: simply try to make the scores as large as possible. Note that this second strategy can only be expected to work reliably in a small area around the origin due to the nonlinearity involved with changing regimes of the process.

Thus a combined strategy is to keep the scores close to the origin, but to look out for the possibility to increase the score values.

3.8.2 Squared Prediction Error

Using the t-score $\mathbf{t}_{new}$ given by equation (3.116) a residual can be calculated

$$\mathbf{E} = \mathbf{X}_{new} - \mathbf{t}_{new}^{\top}\mathbf{P}^{\top}. \tag{3.117}$$

Using this residual the Squared prediction error can calculated in the same way as described in section 3.7.8 which dealt with PCA. The detection and isolation methods are the same too (see page 65).

3.8.3 T_f^2 statistic

3.8.3.1 Detection

The definition of the T_f^2 statistic is based on the on-line t-scores obtained from the process.
The T_f^2 statistic is given by

$$T_f^2 = \mathbf{t}_{new}^\top \mathbf{S}^{-1} \mathbf{t}_{new}, \tag{3.118}$$

where $\mathbf{S}$ is the covariance matrix of the t-scores obtained from the normal data set. Since the t-scores are orthogonal the matrix $\mathbf{S}$ is a diagonal matrix.

The distribution is [Tracy and Young, 1992]

$$T_f^2 \sim \frac{A(I^2-1)}{I(I-A)} F_{A,I-A}. \tag{3.119}$$

3.8.3.2 Isolation

The calculation in equation (3.116) of the t-scores is equivalent to the sum

$$\mathbf{t}_{new} = \sum_{r=1}^{JK} \mathbf{c}_{(r)}^\top \mathbf{x}_{new,r}. \tag{3.120}$$

This sum is changed into a double sum

$$\mathbf{t}_{new} = \sum_{j=1}^{J} \sum_{k=1}^{K} \mathbf{c}_{((k-1)J+j)}^\top \mathbf{x}_{new,(k-1)J+j}. \tag{3.121}$$

The contribution from the jth variable is the corresponding term from the inner sum. The inner sum we will call $\mathbf{t}_j$. This t-score is scaled by means of the covariance matrix to give a y-score

$$\mathbf{y}_j = \mathbf{S}^{-1/2} \mathbf{t}_j. \tag{3.122}$$

The contribution from the jth variable to the T_f^2 statistic can then be calculated as

$$T_{f,j}^2 = \mathbf{y}_j^\top \mathbf{y}_j. \tag{3.123}$$

3.9 Summary

The methodologies described in this chapter are all based upon linear models. These type of models offer the benefit of simple model structure and fast computations. The assumption is that the underlying structure of the system is in fact linear. The estimation of models is fast. Thus, when suitable data are available the model development can be carried out quickly even when some iterations are needed.

The development of the models is in practice an iterative procedure. Several models are often built with the inclusion of different variables, different ways of scaling and different number of components in the model. Using process knowledge and cross validation a low number of useful models can be selected. These models can then be used for their respective applications.

When raw process data first have to be analysed it is recommended that a PCA is performed on the data. This will allow the chemometrician to find patterns in the data. Some of these patterns will be determined by the process. Other patterns will be present only because of faults in the data and these faulty points must be identified before a useful model can be built. When a quality variable is not available PCA can be used for process monitoring. This can be the case in the monitoring of a food product where it is not possible to quantify taste and texture. When a quality variable *is* available it is recommended that a PLS in developed that can take this extra, and very useful, information into account.

3.9.1 Future research

Process chemometrics is an area that is in rapid development. More advanced modelling types are developed and more advanced algorithms for handling existing problems are constructed. Such as handling misssing data, constraints on the parameters, nonlinearities etc. The methods treated in this chapter are all linear and it seems as if future applications of chemometrics will continue to emphasise linear methods, but some development is underway in the direction to include (slightly) nonlinear terms in the modelling [Baffi *et al.*, 1999a,b]. Principal curves is a modelling technique corresponding to a nonlinear version of PCA which also in the future may be useful [Dong and McAvoy, 1996].

The way that batch data is unfolded in order to be suitable for a standard PCA or PLS algorithm is perhaps not optimal. Research in the direction of developing new algorithms for handling the three-way data can take this special structure into account [Bro, 1997]. Application of this modelling structure on spectroscopic data show that new knowledge can be extracted using these methods, but applications in fault diagnosis at this point in time show few advantages [Louwerse and Smilde, 2000; Bro, 1997].

Wavelets used for compression of information or as preprocessing method prior to the chemometric or statistical analysis have been utilised succesfully [Bakshi *et al.*, 1994; Shao *et al.*, 1999]. An introduction to wavelets for chemometricians are given in [Alsberg *et al.*, 1997] and a review of recent applications in chemometrics can be found in [Leung *et al.*, 1998].

List of Symbols

Letters

Symbol	Description
A	Total number of (principal) components in the model.
I	Number of experiments/batches in the data set.
J	Number of measured variables.
K	Number of samples per batch.
Q_α	Critical value for Q.
b_a	PLS model inner relationship parameter.
c_α	Normal deviate cutting off area.
h_0	Used in determining Q-statistic critical value.
l_i	The ith eigenvalue.
$\ln(x)$	Natural logarithm of x.
H_0	Hypothesis.

Matrices

Symbol	Description
$\mathbf{B}$	Regression parameter.
$\mathbf{E}$	Residual matrix.
$\mathbf{R}$	Correlation matrix.
$\mathbf{S}$	Covariance matrix.
$\mathbf{T}$	Scores.
$\mathbf{U}$	Loadings (PCA). Y-scores (PLS).
$\mathbf{V}$	Loadings. $\mathbf{v}_i = \sqrt{l_i}\mathbf{u}_i$.
$\mathbf{W}$	Loadings. $\mathbf{v}_i = \mathbf{u}_i/\sqrt{l_i}$.
$\mathbf{X}$	Data matrix (the independent variable).
$\mathbf{Y}$	Data matrix (the dependent variable).
$\mathbf{Z}$	Scores.
$\mathbf{D}$	Scaling matrix, see equation (B.15)
$\bar{\mathbf{x}}$	Mean vector of $\mathbf{X}$.
$\mathbf{y}_i$	y-scores with unit variance.

Greek Letters

Symbol	Description
Θ_i	Used in determining Q-statistic critical value.
$\kappa(\mathbf{X})$	Condition number of $\mathbf{X}$.
λ	Ridge regression parameter.
σ	Variance.

Symbols

Symbol	Description
$\doteq$	Numerically equivalent to.

4

Supervision of Fed-Batch Fermentations

Process faults may be detected on-line using existing measurements. In the applied approach the modelling is entirely data driven. A multivariate statistical model is developed and used for fault diagnosis of an industrial fed-batch fermentation process. Data from several (25) batches are used to develop a model for cultivation behaviour. This model is validated against 13 data sets and demonstrated to explain a significant amount of variation in the data. The multivariate model may directly be used for process monitoring. With this method faults are detected in real time and the responsible measurements are directly identified. The fault detection and identification is enabled through inspection of a few simple plots. Thus, the presented methodology allows the process operator to actively monitor data from several cultivations simultaneously.

4.1 Introduction

Batch processes are usually very difficult to model due to the circumstances under which they are used. Short runs and large batch-to-batch differences in process conditions make it difficult and time consuming to develop first principles models for the versatile reactor that the batch reactor actually is. Statistical process monitoring (SPM) is commonly used for monitoring continuous processes, where statistical methods are used to monitor that process variables are kept at a stationary level [Pond, 1994]. Fed-batch (or semi-batch) processes are however non-stationary and the process variables are therefore not constant. Thus, it is more difficult to develop a model for normal behaviour and to detect deviations from standard operation.

This paper shows an application of a multivariate statistical method for fault diagnosis. The method uses data which are obtained from existing standard measurements from an industrial process. Hence no measurements have to be added in order to establish the described modelling method than normally would exist in equipment utilized by the fermentation process industries. When

advanced analytical equipment is available (e.g., NIR spectra of the broth) it *can* be included, though. Data is used to develop a model of the normal behaviour of the process. Process knowledge enters the development process when the measurements and the types of batches are specified and selected. The developed model can be used for on-line fault diagnosis and it is also demonstrated that the model can be used for prediction of the product concentration at the end of the batch.

A short introduction to the process described in this paper is given in section 4.2. The crucial part of the process chemometric way of modelling is the availability of process data and as long as this requirement is fulfilled the methods can be used for any process. The data handling is described in section 4.3. The results of applying the methods to a fed-batch fermentation are described in section 4.4 and the paper ends with a discussion and conclusions.

4.2 Process Description

The process investigated in this paper is an industrial fed-batch fermentation process where a *Bacillus* species produces enzymes. In this article the focus is on the main fermentor where the product is produced. The previous steps, spore propagation and seed tank, which have as purpose to produce biomass will not be dealt with, but the methods can be used for those as well to ascertain consistency and error propagation between steps.

The operating procedure for the modelled fed-batch fermentation is to start with a small amount of biomass and substrate in the main fermentor. When most of the initially added substrate has been consumed by the microorganisms the substrate feed is started. This operating procedure is used in order to keep the substrate concentration low during the fermentation. A low substrate concentration in the fermentor is necessary for achieving a high product formation rate due to the catabolite repressor effect.

The fed-batch operating procedure leads to a highly nonlinear process behaviour. Small changes in substrate components or flows can have a large effect on the outcome of the batch and the kinetics of the internal reactions in the cells are subject to limitations. These limitations can be due to diffusion of components or activity of enzymes that both experience S-curve limitations due to changes in substrate concentration. Almost every key variable is changing as the process progresses (volume, biomass, product concentration, etc.). This behaviour distinguishes batch and fed-batch processes from continuous processes where process control can be performed by maintaining key variables constant. In fermentation processes it is costumary to keep the pH and temperature level constant in order to give the cells the best possible conditions for making the desired product. This is also the case for the process described in this paper and the pH and temperature data therefore show very little variation.

The nonlinear behaviour together with limited duration of the process makes it difficult to develop dynamic models of the system. Such modelling requires

detailed knowledge about the microorganisms their metabolism and reaction rates and quantitative data from carefully planned experiments specific to the particular fermentation. Due to the large number of different microorganisms and products used by industry the effort that is needed to develop dynamic models for simulation, fault diagnosis and control seems too large to be overcome in the near future.

Due to the lack of models for model based control the standard operating procedure of fermentation processes is to run with a predetermined feed profile that has been determined as the result of multiple optimization experiments, where high productivity and high reproducibility are the major factors in these experiments. The optimization experiments can be very versatile when trying to optimize operating procedures, substrates and the microorganisms itself, but also takes long time. This is an ongoing task for most processes.

As a result of disturbances and differences in initial conditions the measurement trajectories can deviate from the expected optimal course. If the deviations are not compensated for the batch may have a reduced performance in terms of lower yield and production of unwanted by-products. On the other hand, many upsets of the process that can be seen through changes in the measurements will not have an effect on the quality of the process. In order to detect a fault it is of course necessary that the fault affects, directly or indirectly, the measurements.

Abrupt, gross faults in single variables can easily and reliably be detected by a conventional process control system. Drift of variables and faults involving multiple variables are not easily detected. These more complicated faults, even if they are small, can have a large effect on the quality of the process; an effect that is difficult to predict without advanced tools.

It is conventionally up to the process operator to determine when deviations are unacceptably large and will have an effect on the product quality and the productivity. The reliability of this highly manual procedure depends on the training that the process operator has received, his experience and the character and number of processes that he has to supervise simultaneously.

The aim of the methods presented in this paper is to provide the process operators with a tool for detection and isolation of faults by limiting the amount of data that the process operator has to monitor in order to evaluate the present and future operation of the process. This tool is especially beneficial when process operators are monitoring several processes at the same time.

4.3 Data Analysis

A set of on-line measurements are obtained from the process at regular sample intervals. These measurements have been stored for many past fermentations forming a database of historical information about the process. The measurements available for the considered process are shown in table 4.1. Note that all the measurements are unsophisticated standard measurements. Thus, no ad-

vanced or expensive measurement devices has to be installed in order to make the methods work; the particular type of measurement is not important as long as the measurements represent the state of the system.

The model is entirely data based as opposed to conventional first principles modelling. It is up the modeller to choose data in a way such that the data describes the behaviour of the system. The selection of the correct data for the modelling work is the most important step of the modelling phase. When a process data base already exists the task is reduced to selection among the data. Better results can usually be obtained if the data are obtained from designed experiments, but this is costly and time consuming and is not considered in this paper.

The data used in this paper are obtained from a historical data base. It is desired that the model will work for all future batches. Therefore a model is built using all available data sets (except for the data sets used for validation). A few of these data sets had to be left out because the course of the batches are so incompatible with the other batches that a single model cannot be formed that includes the variation of all the batches. The nonconforming data are left out in order to develop a model of the desired behaviour of the process. When a new batch is monitored one can then see if the batch is operating within the window of the desired behaviour given by the model. If the batch deviates too much one can say that a fault has occurred and that some action must be taken. As the model has been built on historical data where process faults have not been treated as suggested in this paper some faults are unfortunately allowed to persist. It is the goal that once fault diagnosis has been implemented and the process variation has decreased a new and better model will be developed leading to an ever improving process as this iteration progresses.

Other types of models can be developed that describe more specialised types of behaviour. Separate models can be built using data from batches that have a given high or low yield. Models can be built using data from batches that have experienced a certain type of fault. Building such specialized models gives higher specificity towards the type of error, but at the same time the specificity towards previously unseen errors is diminished.

Data from the batch and fed-batch processes can conveniently be put in a three-way matrix $\underline{\mathbf{X}}$ $(I \times J \times K)$. I is the number of batches, J is the number of variables (10), and K is the number of samples from each batch (114). The numbers in parentheses refer the the numbers actually used in this paper. The

Table 4.1. On-line measurements obtained from the fermentation process.

1	Total amount of substrate	2	Agitator power input
3	Total amount of antifoam	4	Weight
5	pH	6	Temperature
7	Dissolved oxygen	8	Air flow
9	CO_2 % in off-gas	10	O_2 % in off-gas

size can vary by orders of magnitude depending on the process duration and available measurements when other processes are modelled.

The matrix $\underline{\mathbf{X}}$ can be unfolded to a two-way matrix, see figure 4.1. This two-way matrix is called $\mathbf{X}$ $(I \times KJ)$. For each fermentation (a row in $\mathbf{X}$) a quality measure is recorded and stored in a matrix $\mathbf{Y}$. The measure used here will be the final product concentration, but one could also use, e.g., the productivity. Each column of $\mathbf{X}$ corresponds to a certain variable at a certain point in time. If the process is carried out following a predetermined feed profile it is expected that the trajectories of the measurements are similar and that the mean value of a variable at a certain point in time can be used as a reference value for future processes. The goal of the monitoring is to observe and minimize deviations from this reference value in future batches. Thus, to facilitate the analysis the columns are centred and scaled to unit variance.

The matrix $\mathbf{X}$ is rather large, but the columns of $\mathbf{X}$ are not independent. They describe similar events in the process and the dimension of the space spanned by $\mathbf{X}$ is usually very low. Thus, by using a multivariate statistical technique to reduce the dimensionality of the variable space the problem of describing the process becomes simpler to handle. Principal Component Analysis (PCA) is frequently used for this purpose and is recommended if no quality variables are available, which is frequently the case for many biotechnological processes. When quality variables are available one can use Principal Component Regression (PCR) or preferably Projection to Latent Structures (PLS) which is a linear regression method that optimally utilizes the information in $\mathbf{X}$ and $\mathbf{Y}$ at the same time [Jackson, 1991].

When the variable space is compressed using either PCA or PLS the process can be monitored in a low dimensional space using simple plots [Nomikos and MacGregor, 1995; Nomikos, 1995]. The modelling methods requires data from several batches in order to produce a good model. Sometimes 20 batches are sufficient, but 50 or even 100 can be necessary if the dimensionality of the model space is large. The number can be reduced if some experiments have been carried out that fills the model space in an efficient way, but this is rarely a possibility for production scale fermentations.
Here a quality variable is available for the described process and therefore a PLS model will be developed. PLS is defined by a bilinear model that is used to model the relationship between $\mathbf{X}$ and $\mathbf{Y}$

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{W}^{\top} + \mathbf{E} \\ \mathbf{Y} &= \mathbf{U}\mathbf{Q}^{\top} + \mathbf{F} \end{aligned} \qquad \mathbf{u}_a = b_a \mathbf{t}_a. \tag{4.1}$$

PLS maximizes the covariance between $\mathbf{u}_a$ and $\mathbf{t}_a$ and the number of components A (number of columns in $\mathbf{T}$) is chosen such that $\mathbf{E}$ and $\mathbf{F}$ are small in some sense. The data are in other words reduced to a number of scores (either $\mathbf{T}$ or $\mathbf{U}$) that lie in a low dimensional space of the data, but describes a good deal of the variation of the data. The expressions in (4.1) can be rewritten as

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{F}^{*}. \tag{4.2}$$

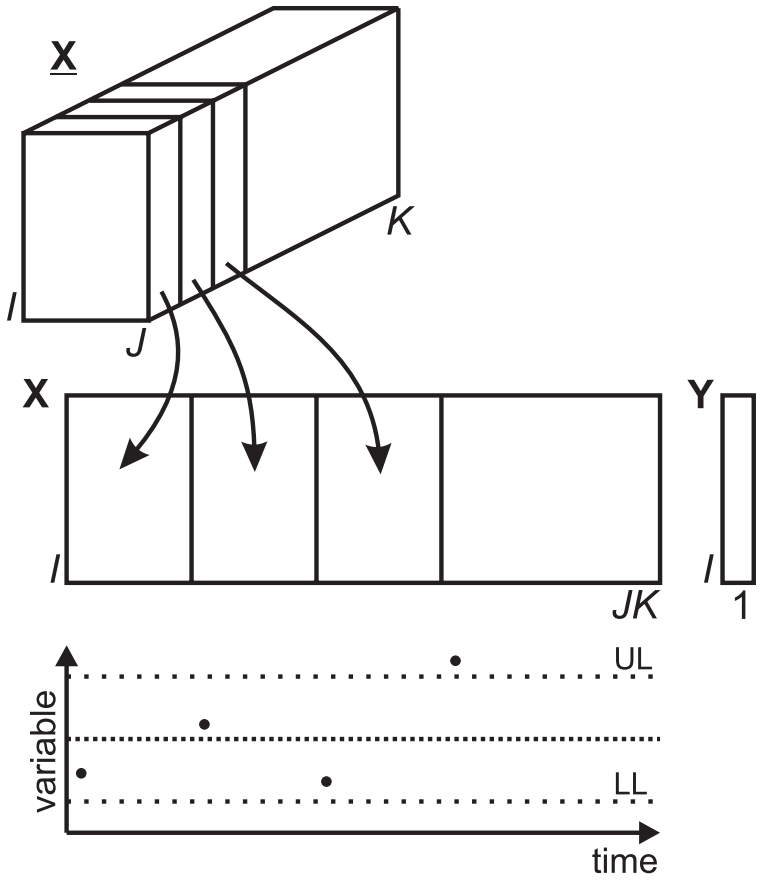


Figure 4.1. Unfolding of a three-way matrix to form a two-way matrix. **X** contains the on-line variables and **Y** some measure of the quality of the process (here: the final product concentration). The principle behind process chemometrics is shown in the lower part of the figure for a single variable. Every time a new measurement is obtained it will be compared to the expected level. If the deviation is too large (above upper (UL) or below lower (LL) limits) the process is behaving abnormally and the process operator should take action.

This expression is in many cases is easier to work with. The regression parameter matrix is given by

$$\mathbf{B} = \mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1}\mathbf{Q}^\top, \tag{4.3}$$

where the loading matrices $\mathbf{W}$, $\mathbf{P}$ and $\mathbf{Q}$ are determined by the PLS algorithm [Höskuldsson, 1996; Martens and Næs, 1989]. $\mathbf{Y}$ can be predicted using:

$$\mathbf{Y} = \mathbf{XB} = \mathbf{XW}(\mathbf{P}^\top\mathbf{W})^{-1}\mathbf{Q}^\top \tag{4.4}$$

$$= \mathbf{TQ}^\top. \tag{4.5}$$

The model can be used for calculating a vector $\mathbf{t}$ (a t-score) for a new data set $\mathbf{X}_{new}$

$$\mathbf{t}_{new} = (\mathbf{X}_{new}\mathbf{W}(\mathbf{P}^\top\mathbf{W})^{-1})^\top. \tag{4.6}$$

This expression can be used only when all data from a fermentation is available. For on-line purposes a full $\mathbf{X}$ matrix has to be constructed. In this paper $\mathbf{X}$ will be constructed by using all of the available information collected up to the current time and the remaining part of $\mathbf{X}$ will be filled with a copy of the most recently obtained measurement. This results in good fault detection properties and reasonably well behaved estimation of $\mathbf{Y}$ as well. This way of filling $\mathbf{X}$ corresponds to predicting what would happen if a fault is allowed to remain unchanged for the remaining duration of the batch and is a way to evaluate the seriousness of faults. This procedure is justified because the process dynamics become increasingly slower as the tank is filled and less change of the concentration variables is observed especially as the product concentration stabilizes during the last part of the batch.

4.3.1 Fault Diagnosis

Fault diagnosis consists of three steps: Fault detection, isolation and identification (FDII). The methods presented in this section will readily detect faults and isolate the measurements that are behaving abnormally and the methods may facilitate the identification of the fault, i.e. to determine the physical origin of the fault in the process.

For detection two statistics, the T_f^2 and the standard prediction error, can be calculated The T_f^2 statistic (based on the Hotelling T^2 statistic [Mardia *et al.*, 1995]) is calculated using the scores

$$T_f^2 = \mathbf{t}_{new}^\top\mathbf{S}^{-1}\mathbf{t}_{new} \sim \frac{A(I^2-1)}{I(I-A)}F_{A,I-A}, \tag{4.7}$$

where $\mathbf{S}$ is the covariance matrix of the t-scores contained in the matrix $\mathbf{T}$ calculated during the model development [Tracy and Young, 1992], I is the number of batches used for modelling and A is the number of components. F denotes the F distribution.

The squared prediction error (SPE) is calculated by

$$SPE_k = \sum_{r=(k-1)J+1}^{Jk} \mathbf{e}_{(r)}^2, \tag{4.8}$$

where $\mathbf{e}_{(r)}$ is the rth column in the matrix $\mathbf{E} = \mathbf{X}_{new} - \mathbf{t}_{new}\mathbf{P}^\top$. The distribution of the SPE can be approximated by a weighted χ^2 distribution $SPE_k \sim (v_k/2m_k)\chi^2_{2m_k^2/v_k}$, where m_k and v_k are the mean and variance of the SPE obtained for the data set used for the model development at time instant k [Nomikos, 1995].

A fault is detected whenever the T_f^2 statistic or the SPE exceeds, e.g., a 95% confidence limit. The 95% limit is usually taken to be a warning level only and action is taken when the statistic exceedes a 99% limit. The T_f^2 statistic reveals faults that can be described by the model. The SPE will show if a totally new event is occurring in the process. This includes unusual variation of the controlled variables stabilized by simple control.

The process can also be monitored using the scores in a socalled score plot. Usually the number of components is low (2–3) and therefore a single plot is usually sufficient to display the state of the process. If the model contains more than 2 components one can either construct three-dimensional plots or make several two-dimensional plots to show the variation. Confidence limits can be established using the ellipsis defined by (4.7). These limits are suitable when developing the model since the data used in the development are all from similar batches (in some sense). When the model has been based on only a few data sets it is often seen that the ellipses are not filled evenly. This can be due to vialation of the normallity assumptions of the scores or simply a result of basing the model on historic data that do not fill the entire score space because the data were obtained under normal operationg conditions. It has been proposed that kernel density estimates of the confidence limits can be beneficial when monitoring the process [Martin *et al.*, 1996]. Using kernel density estimates for the limits it is assured that we only put confidence in scores that are in area that we have actually seen before when building the model.

An general estimator for the kernel density can be defined as [Simonoff, 1996]

$$f(\mathbf{t}) = \frac{1}{n|\mathbf{H}|} \sum_{i=1}^{I} K_d\left(H^{-1}(\mathbf{t} - \mathbf{t}_i)\right), \tag{4.9}$$

where $|\mathbf{H}|$ is the absolute value of the determinant of the matrix $\mathbf{H}$, which is a bandwidth matrix. K_d is the multivariate kernel function. One way of creating K_d from a univariate kernel K is by using a product kernel

$$K_d(\mathbf{u}) = \prod_{j=1}^{d} K(u_j). \tag{4.10}$$

The Gaussian kernel $K(u) = (2\pi)^{-1/2}\mathrm{e}^{-u^2/2}$ will be used here. Other kernels may be used, but the resulting plots do not vary much when other kernels are chosen. The bandwidth matrix is chosen to be a diagonal matrix. This complies with the scores being independent. The bandwidth has to be chosen based on the shape of the data and the coarseness that is desired in the plot. A default value for the bandwidth is $h_i = 1.059 s_i I^{-1/5}$, where s_i is the standard deviation of the ith score [Simonoff, 1996]. For confidence limits in the score plots it has been found that the bandwidth usually has to be sligtly larger than the default value leading to a coarse confidence region.

4.4 Experimental Results

A model has been developed by carefully selecting data sets from the historical data base that reflect the normal desired operation of the fermentations. This has been done by first discarding any batch that has very large undesired or unusual behaviour compared to the desired batch behaviour (e.g. because of experiments or infections). When a batch is very short or very long it is discarded, too. The data suitable for the model development is truncated such that 114 time samples are included in the model. As mentioned the dynamics of the process become slower towards the end of the batch and there are usually few corrective measures to be performed if a fault occur near the end of a batch, anyway. An initial model is estimated and batches are removed from the modelling data set if they are lying outside a 99% confidence bound in a score plot using ellipses as confidence bounds. The procedure is carried out iteratively until all batches remain within the 99% confidence bound. It is important here to identify *why* a batch does not lie within the confidence bound in order to make sure that only batches that are really not conforming are eliminated from the normal data set.

After the reduction in the number of data sets 25 data sets were used for model development and 13 for validation of the model. Using the prediction error sum of of squares (PRESS) as validation criterion 2 components are found to be sufficient for describing the relationship between $\mathbf{X}$ and $\mathbf{Y}$. The obtained model uses only 28% of the information in $\mathbf{X}$, but explains 80% of the variation of $\mathbf{Y}$. The low percentage of used variation in $\mathbf{X}$ is due to the inclusion of controlled variables that have low variation (e.g. pH and temperature). These variables are *known* to have large influence on the product formation and that is the reason they are controlled. If the influence of the controlled variables on the product formation is to be modelled by the PLS model these variables must be perturbed in designed experiments. These experiments have not been performed since this is expected to lead to a decrease in product formation. If the PLS model was to be used entirely for prediction purposes and it is assumed that the control is perfect the model performance could be improved by not including the controlled variables in the model. In the present case we are interested in fault diagnosis of the controlled variables, too, and therefore

leave the controlled variables in the model.

4.4.1 On-line estimation of Final Product Concentration

Using equations (4.4) to (4.6) the final product concentration can be predicted in real time. Figure 4.2 shows the performance of this method for a low producing fermentation. The data from this batch, which was used for model validation, but not for the modelling itself, will be used in the following sections. The final product concentration is 0.40 whereas the average value for the batches used for model development is 0.46. Figure 4.2 shows that the model is able to predict the final product concentration within 10% during most of the fermentation except during the interval from 70 to 80, where there is a large deviation due to a process fault. The accuracy of the estimation is only slightly larger than the accuracy obtained by the lab when chemical analyses are performed. This deviation can be interpreted further by the fault diagnosis as described in the following section.

The T_f^2 statistic in figure 4.3 plotted as a function of time for the same batch as in figure 4.2, shows that this particular process has a large deviation from $t = 70$ to $t = 80$. The slow drift of the T_f^2 that can be noted is difficult to detect

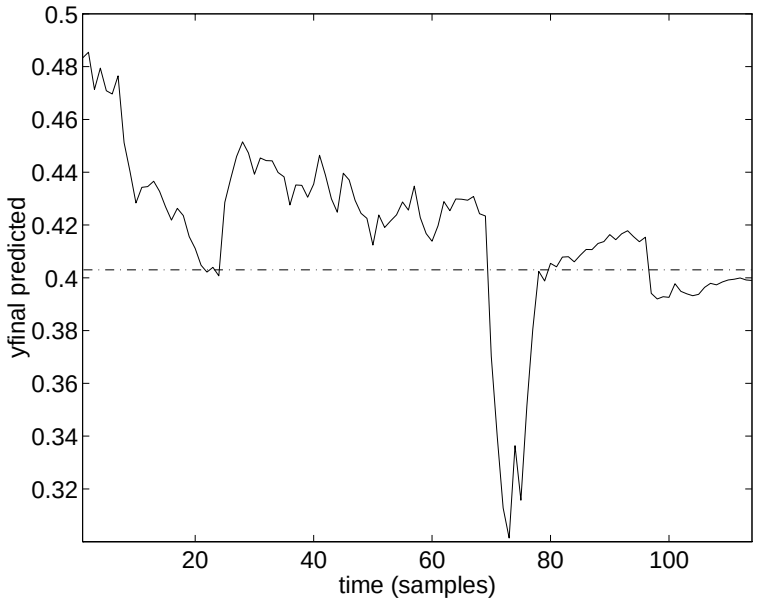


Figure 4.2. Prediction of final product concentration. The dotted line indicates the actual product concentration as it was measured at the end of the batch. If the large fault at $t = 70$ was allowed to persist throughout the fermentation the product concentration was estimated to be much lower than the one actually obtained.

looking at the raw measurements. It has already been shown (in figure 4.2) that this process drift results in a much lower than average product concentration at the end of the fermentation. Figure 4.2 illustrates the importance of this type of monitoring to predict the consequences of deviations.

The SPE in figure 4.4 indicates that this process is deviating from the average process almost throughout the entire fermentation.

Contribution plots, which indicate the variables that are contributing the most to the T_f^2 statistic or the SPE, can easily be constructed and can thus be used in the fault identification [Miller *et al.*, 1993; Nomikos, 1995]. Contribution plots can either be used to find the change in the contribution from one point in time to another or the contribution plot can be used to find the deviation of the current batch when compared to the normal batch behaviour described by the model. We here choose to look at the fault which has been detected around $t = 70$. Figure 4.5 shows the change in contribution of the variables from a point in time just before the fault could be detected in the SPE and T_f^2 plots ($t = 67$) to the point where the fault is at its highest ($t = 73$). The figure shows that there is a large change in the contribution of the CO_2 and O_2 measurements. Thus the task of isolating faulty measurement has been reduced to looking only at a few plots that show that a fault has occurred and contribution plots to identify which variables that contribute to the fault.

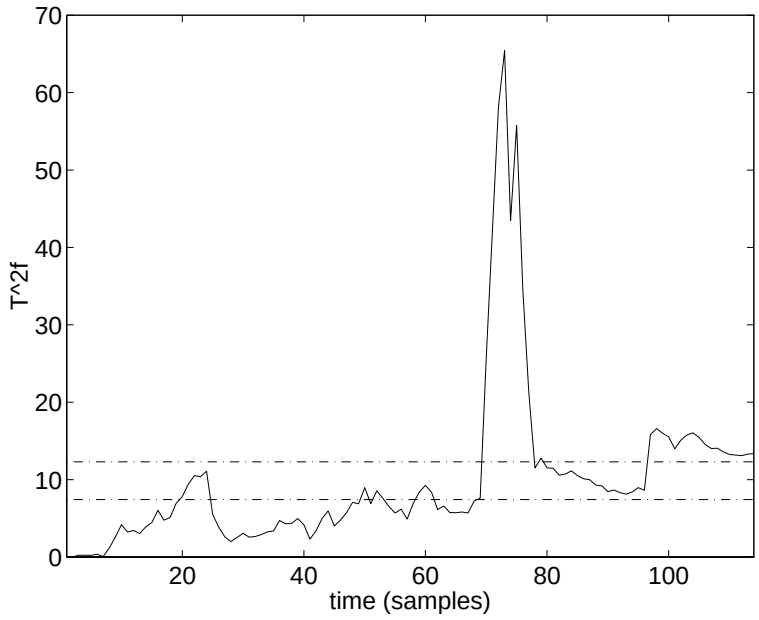


Figure 4.3. T_f^2 statistic. Dash-dotted lines indicate 95% and 99% confidence limits.

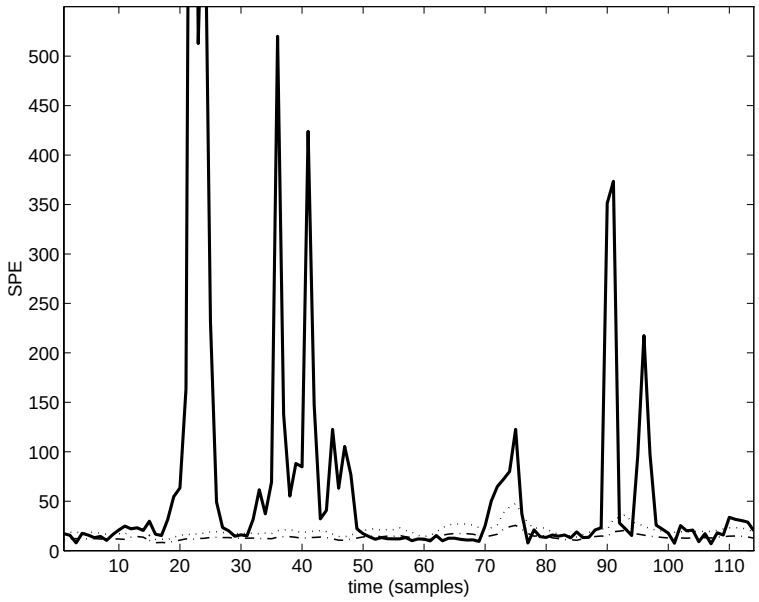


Figure 4.4. Squared Prediction Error (SPE). Max value at peak when $t = 20$ is about 1400. The dash-dotted line is the 95% and the dotted line is the 99% confidence limits.

4.4.2 Score plots

A score plot can be used to monitor the process. Since the model only contains 2 components the plot in figure 4.6 is the only one needed to monitor the major variations of a normally operating batch. The figure shows the variation of the process in the reduced space of the two components.

The score plot describes the present state of the process and allows the operator to interpret the development of the process. Emphasis must be put on the word *interpret* because the scores usually lack any direct physical meaning. One way of interpreting the score plot is to ascribe different phenomena to the movement of scores. E.g., this model shows large variation of the t_2 score when deviations in the O_2 uptake and CO_2 production occur. Another way of finding a physical relationship is to investigate the loading matrices, which directly show the relationship between the measurements and the scores. Both interpretations can be useful when the behaviour of a batch is to be described and current or future faults are to be eliminated.

Equation (4.5) shows the relationship between the scores and the dependent variable y as $\mathbf{Y} = \mathbf{T}\mathbf{Q}^{\mathsf{T}}$. Since $\mathbf{Q} = [\,0.05\ 0.08\,]$ in this case it can be inferred that batches will have a higher than average product concentration at the end of the batch when the score values all are positive.

Score plots can furthermore be used as a fingerprint of the batch. Instead of

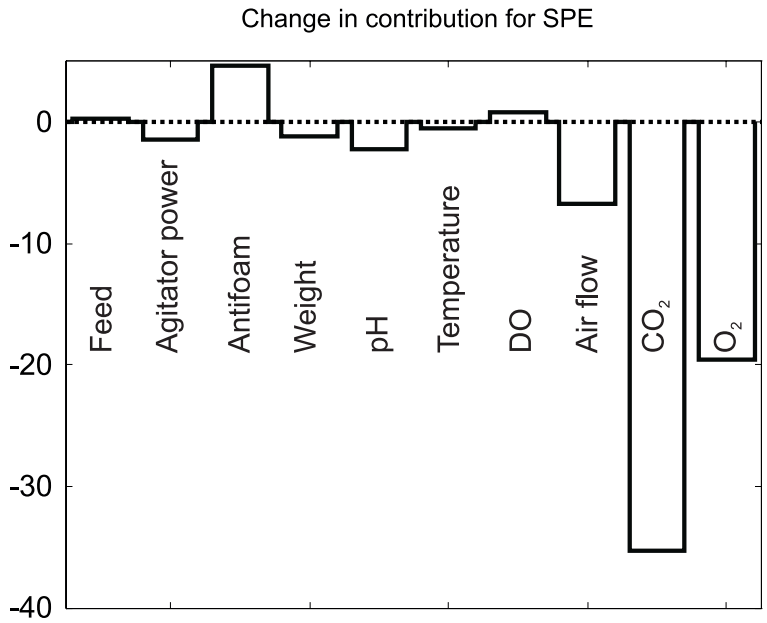


Figure 4.5. Contribution plot showing the change of the process from $t = 67$ to $t = 73$. It is seen that the variables CO_2 and O_2 give highest contribution to the fault and that they are lower than normal. From the plot of the predicted final product concentration (figure 4.2 it can be seen that the fault has a negative effect on the quality.

looking at a plot of the different measured variables one can look at a score plot for an entire batch to investigate if something unusual has happened during the fermentation. Figure 4.7 shows such a score plot of a well behaved batch. The scores stay in this figure close to the point (0,0) which shows us that this batch did not have any faults that affected the product concentration. It would have been much harder to interpret the original measurements due to their time varying nature.

4.5 Discussion and Conclusion

The methods shown above are powerful tools for compressing and displaying process information in a meaningful way. The methods can be used both for fault diagnosis and for prediction purposes. It should be noticed that the predictions are obtained in real time as opposed to wet chemical analyses—with almost the same accuracy.

Using the demonstrated methods the operator is provided with a clear view of the process performance. Instead of watching 10 (correlated) variables at the same time it is sufficient to inspect only two simple plots in order to evaluate

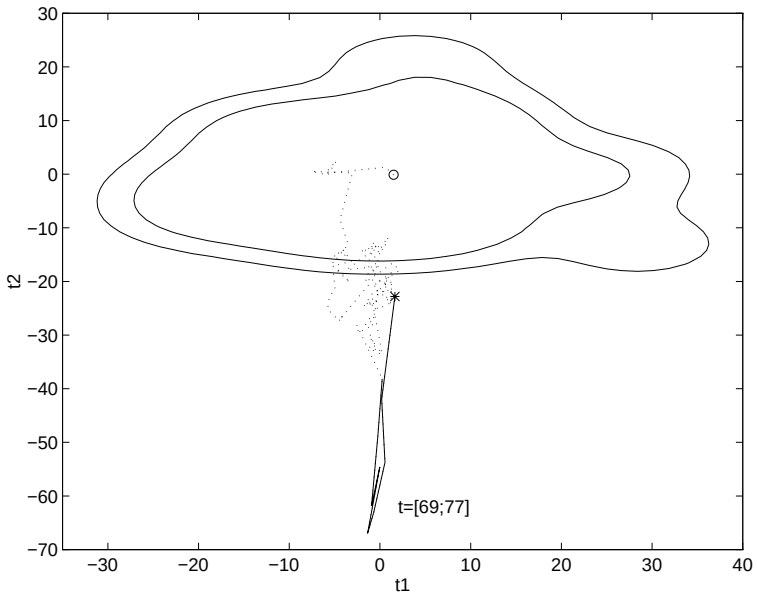


Figure 4.6. Score plot for a faulty batch illustrating the development of the process in a reduced space. The beginning of the fermentation is marked with a "$\circ$". The time interval [69;77] (starting with a "$*$") are shown as solid lines. The score t_1 varies mainly when there are large oscillations in the temperature. t_2 varies mainly when the CO_2 and O_2 measurements change.

the present and future behaviour of the process.

The relationship between the measurements and the quality variable (product concentration) is utilized by the model such that measurement deviations that do not signify a quality change are not marked as fault in the T_f^2 and score plots. If quality variables are unavailable or it is believed that any measurement deviation should be marked as a fault a process model using PCA instead of PLS may be developed. Such a model will lead to the same type of fault diagnosis plots.

The displayed figures are intended as process operator tools to ficilitate monitoring process performance. With these tools operator attention can be directed mainly at faulty processes instead of having to constantly watch all concurrently running processes.

As displayed here these data-driven methods are only used for supervision and not directly for control. It will be a relevant future step to develop an expert system that can be used to interpret the score plots and automatically take appropriate action when certain kinds of (known) faults occur. Without any doubt, the described chemometric tools will lead to a higher utilization of process data in modelling, optimization and control of complicated processes because process chemometrics provide the process industries with data handling

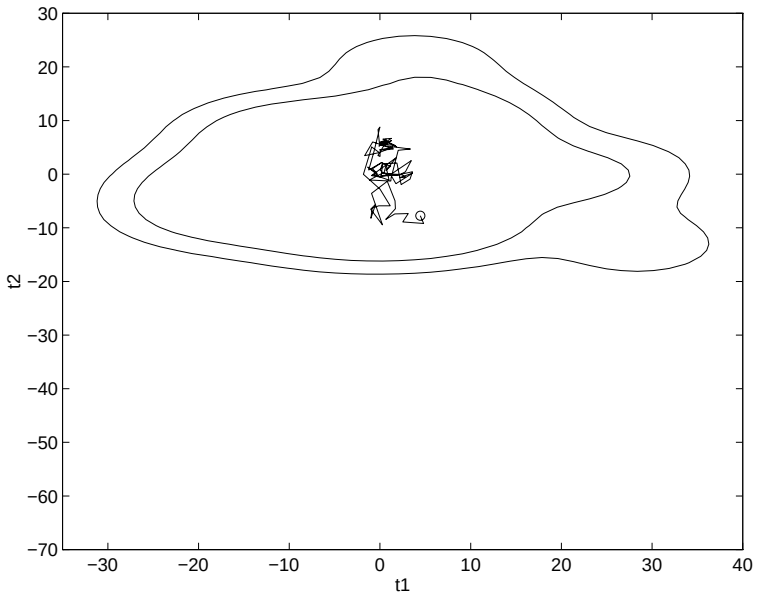


Figure 4.7. Score plot for a well behaved batch. The scores remain near the point (0,0) suggesting that the product concentration will end on the average value of the batches that where used to form the model.

methods that can utilize the process knowledge contained in process data, which currently are obtained and stored for many processes.

5

Dynamic I/O Modelling for Batch Processes

Dynamic linear time invariant (LTI) models constitute an important model type which often is applied in process control and dynamic optimisation. This linear model type is important because it is straightforward to derive the models (from e.g. data or nonlinear models) and to apply the models for process control and process optimisation. Linear models can be used for local analysis of the underlying plant in many cases even when the process is nonlinear. Batch processes are often described using nonlinear and/or time-varying models, but their behaviour may be approximated with a set of local linear models.

This chapter describes methods for modelling a batch process operating near a reference trajectory as a set of linear time invariant models. This set of models is combined into a single model where all parameters are estimated simultaneously by stacking time shifted models on top of each other in order to cover the entire batch. This model type is called a stacked state space model. When the models are estimated it is essential to apply regularisation methods to the problem in order to estimate the many parameters with high accuracy.

Fed-batch processes often used in chemical and biochemical processes can be modelled using first principles models, but this model type can be very time consuming to develop for industrial scale processes. For most industrial scale processes a large number of measurements are obtained at regular intervals. These measurements describe the system behaviour to some extent and can form a basis for a model of the system. This is done by modelling the system using input/output time series models. These models are powerful modelling tools and exist in linear and nonlinear versions. The focus will only be on linear models because they are easily adapted to noisy industrial data and have a large methodogical basis for system analysis and optimisation procedures. Since batch processes change dynamics with time, i.e. are time varying, the developed linear models can be viewed as a collection of linear models where a new linear model is chosen at each sample time. Note however, that all linear model parameters must be estimated simultaneously in order to exploit the full potential of the proposed modelling paradigm.

The developed models can directly be used for monitoring and control purposes as it be shown in the following chapters. Furthermore the developed models can be used in the batch to batch control scheme used by [Lee *et al.*, 1997].

The models derived in the chapter can be viewed as linear state space models. This is a model type that has been dealt with in great detail by control engineers for decades. An introduction to state space models can be found in [Rugh, 1996]. System identification is also an important part of this chapter. A review of continuous-time identification is given in [Unbehauen and Rao, 1998]. Ljung presents the classical introduction to discrete linear time-invariant and time-varying input/output modelling in [Ljung, 1987].

Modelling of batch processes using input/output models is not covered in the literature in great detail. However, some information can be found in [Russell *et al.*, 1998]. Masses of literature on I/O modelling of time-varying systems do exist, but batch processes are not dealt with as a special case [Dewilde and van der Veen, 1998]. The use of subspace identification methods on time-varying systems is covered in [Verhaegen and Yu, 1995; Liu, 1997]. The use of Partial Least Squares (PLS) and Canonical Variate Analysis (CVA) for modelling continuous processes is treated in [Simoglou *et al.*, 1999; Negiz and Çinar, 1998; Çinar and Undey, 1999]. Although the literature that deals with identification of time invariant and time variant systems is a growing field with many new and diverse methods there is generally very little written about identification of input/output models for batch processes.

The work in [Lee *et al.*, 1997; Chin *et al.*, 1998] is based on linear models for batch-to-batch optimisation. In these papers linear models are used to improve the batch-to-batch behaviour using control. The approach by Lee *et al.* is not investigated in this chapter, but it must be noted that the model type and model estimation procedures presented there are *directly* suitable for use in the described control and optimisation algorithms, but [Lee *et al.*, 1997; Chin *et al.*, 1998] allocate little space to the derivation and identification of the dynamic models used.

Reviews of control of batch processes in general in given in [Berber, 1996]. The control of fermentors is reviewed in [Rani and Rao, 1999; Shimizu, 1993].

The structure of the chapter is as follows. Section 5.1 motivates the need for dynamic modelling of batch processes. The dynamic model type used will be based on an input/output ARX-model described in section 5.2. This model type can be applied—as it is—only to continuous processes therefore an adaptation to batch processes is necessary and will be introduced in section 5.3. Estimation of parameters in the model for the single input/single output (SISO) case is described in section 5.4 for batch processes and an validation example is given. Methods for simulation are introduced in section 5.5. Batch processes are multivariate therefore is is necessary to expend the models presented until this point to multi input/multi output (MIMO) systems. The handling of the data and the algorithms is the same as for the SISO case, but the models do

increase in size and complexity. The MIMO models are presented in section 5.6. Sections and 5.8 ends the chapter with discussion and conclusions.

5.1 Dynamic Models

The goal of this chapter is to develop methods for modelling batch processes from data using linear time invariant models (LTI models). This can be seen as a very challenging task since batch processes conventionally are modelled as nonlinear ordinary differential equations (ODE), which lead to time-varying models when linearised since there is no fixed operating point. The model structure proposed in this chapter is based on assembling a set of local linear models into one single large model structure. The local linear models are obtained as a representation of the dynamics at each sample instant and a swicth is made between the local models at each sample instant.

Models for batch processes are most commonly built using nonlinear modelling methods for continuous system adapted to batch processes. When the models are used for process optimisation the optimality conditions are based upon the final state of the system. Thus a high accuracy over a long time span is therefore desirable. Special handling of the start and end of the batch is required and similarly for events occurring during the batch where large changes to the process may be introduced. Attention at these points during the batch are necessary as it is at these points in time that the operation of the batch may have the largest effect on the course of the batch. The prevailing operating strategy in today's batch processing plants is to operate the batch processes using recipes. This operating strategy means that batches are repeated many times with almost the same trajectories, which must also be utilised by the model structure.

The following section will give a short introduction to dynamic input/output modelling with special attention to the notation and model types that will be developed in subsequent sections in this chapter.

5.2 ARX models

Input/output models cover a broad range of model types that maps measurements of manipulable and disturbance variables and output variables. The most simple linear model structure for this kind of modelling is the AutoRegressive with eXogeneous input model (ARX). Input to the system is denoted $u(t)$. The output is denoted $y(t)$. Both inputs and outputs may be multivariate. In that case the $u(t)$ and $y(t)$ are vectors instead of scalars; not necessarily of the same length. In the following sections only the single input/single output case will be shown to keep the presentation simple. A list of symbols is given at the end of this chapter on page 126.

Previous measurements of the inputs and outputs are used in an ARX model

to predict a future output at time $t = kT_s$, where T_s is the (constant) sample time:

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + \cdots + a_{n_a} y(k-n_a) + b_1 u(k-1) + b_2 u(k-2) + \cdots + b_{n_b} u(k-n_b). \quad (5.1)$$

The coefficients a_i and b_i are constants so this model type is a linear time invariant model. The coefficients are determined by an identification scheme where data obtained from the system is used in an estimation procedure.

The outputs $y(t)$ and inputs $u(t)$ in the ARX model are deviation variables. This means that a reference value $\bar{y}$ (often the mean) is subtracted from the measured value before it enters the model. This simplifies the model by eliminating a constant term and enhances numerical properties of estimating the model parameters since the absolute values of the data become smaller. One can furthermore scale the outputs and inputs to adjust for differences in range or variance.

In the remaining part of this chapter the calculations are performed in deviation variables. For some calculations it is necessary to use the relationship between the real variables $\mathcal{Y}(t)$ and the deviation variables $y(t)$

$$\mathcal{Y}(t) = \bar{y} + y(t). \quad (5.2)$$

In the case where we have a batch process it is possible to make deviation variables not just around the total mean of the output, but it is possible to use the mean of the process at a particular time as the basis for the calculation

$$\mathcal{Y}(t) = \bar{y}(t) + y(t). \quad (5.3)$$

Similar equations can be set up for $\mathcal{X}(t)$ and $\mathcal{U}(t)$ that are the physical variables for the states and inputs, respectively.

A range of time series and input/output model types exist. Many of these are described in [Ljung, 1987; Box *et al.*, 1994]. The ARX model is chosen for its simplicity. It is simple in structure and for estimation of the parameters. It will of course be a topic for further investigation to observe how other modelling types (linear, bi-linear and nonlinear) with different noise models can be tailored to the framework presented in this chapter.

5.3 Stacked State Space Models for Batch Processes

A LTI-model of a batch process can be built if it is possible to linearise the model around a normal operating trajectory. If the variations of the process around the trajectory are not too large a linear model will give satisfactory accuracy. This is often the case for industrial batch processes. In order to

make LTI models for batch processes an input/output formulation is developed. A new state vector $\mathbf{x}$ is defined. This vector contains all measurements from the entire batch. This assumes equal duration of the batches which is also a common goal in industrial batch processing. The vector $\mathbf{x}$ can thus be seen as a state vector for the entire batch system and is defined as

$$\mathbf{x} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}, \tag{5.4}$$

where y_i is $y(t = T_s i) = y(k = i)$. The sample time is assumed constant. The developed methodology allows, however, the sample time to be non constant as long as samples are made at the same time instant for all batches. It can be convenient to have more samples during time intervals where the process experiences large, fast changes and less samples when little happens to the process. The sample time will be left out of the following equations to avoid notational complexity.

A simple model can be made that relates the input directly to the measured output:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \mathbf{B} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{n-1} \end{bmatrix}, \tag{5.5}$$

where $\mathbf{B}$ is a square $(n \times n)$ matrix. A more simple notation for this model is:

$$\mathbf{x}_k = \mathbf{B}\mathbf{u}_{k-1}, \tag{5.6}$$

where $\mathbf{x}_k$ contains the n measurements from $k = 1$ to $k = n$ and $\mathbf{u}_{k-1}$ contains the n input values from $k = 0$ to $k = n - 1$. A similar model type is also used in [Russell *et al.*, 1998] for a batch process, but in that paper only a static model is formed and only a terminal quality variable is included on the left hand side.

The model (5.6) assumes that the batch is initiated with the same conditions for all batches. Even when this is attempted, it is not always achieved and a more advanced model has to be used e.g. by including a component for modelling the initial conditions and/or estimates of the input trajectory during the batch. In the following past outputs as well as inputs are used to estimate new outputs.

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}, \tag{5.7}$$

where $\mathbf{A}$ and $\mathbf{B}$ are square matrices with dimension n. This is an ARX type of model with one $\mathbf{A}$ matrix and hence it can be viewed as a state space model where measurements obtained at different times are stacked into a state vector. Thus, the model that has been created is a finite-horizon, time-shifted, stacked, linear state space model, which will just be called a stacked state space model.

5.4 Parameter Estimation

It is assumed that data are collected from several batch runs. These batches should be performed using different initial conditions and different input trajectories during each of the batches in order to exite the system [Ljung, 1987]. The output values are collected in the matrices $\mathbf{X}$ $(n+1 \times I)$ and $\mathbf{U}$ $(n+1 \times I)$, where I is the number of batch runs. Column i of the matrices contains data from the ith batch. The number of batch runs will usually be (much) less than the number of sampled data points. For modelling purposes the matrices $\mathbf{X}_k$ and $\mathbf{U}_{k-1}$ are defined that contain the vectors $\mathbf{x}_k$ and $\mathbf{u}_{k-1}$ from different batches, respectively.

For estimation of the parameters in the simple model $\mathbf{x}_k = \mathbf{B}\mathbf{u}_{k-1}$ the following least squares problem is formulated.

$$\min_{\mathbf{B}} \|\mathbf{X}_k - \mathbf{B}\mathbf{U}_{k-1}\|. \tag{5.8}$$

A solution to this problem can be found using

$$\mathbf{B} = \mathbf{X}_k \mathbf{U}_{k-1}^{\#}, \tag{5.9}$$

where $^{\#}$ denotes the pseudo inverse of a matrix, which is most accurately calculated using the Singular Value Decomposition (SVD) [Golub and van Loan, 1991].

For finding the parameters in the model $\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{B}\mathbf{U}_{k-1}$ we rewrite the equations into

$$\mathbf{X}_k = \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix} \begin{bmatrix}\mathbf{X}_{k-1} \\ \mathbf{U}_{k-1}\end{bmatrix}, \tag{5.10}$$

The solution to this problem can be stated as a least squares problem

$$\min_{\begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}} \left\| \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix} \begin{bmatrix}\mathbf{X}_{k-1} \\ \mathbf{U}_{k-1}\end{bmatrix} - \mathbf{X}_k \right\|. \tag{5.11}$$

A solution to this problem can be found again using the pseudo inverse

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix} = \mathbf{X}_k \begin{bmatrix}\mathbf{X}_{k-1} \\ \mathbf{U}_{k-1}\end{bmatrix}^{\#}. \tag{5.12}$$

5.4.1 SISO Example

A small single input/single output example is here given in order to illustrate the capabilities of the proposed models. The results of the example should be obvious from the description in the previous sections, but are included here in order to prepare for MIMO modelling in section 5.6. A simple SISO system is used in the example, which does *not* model a physical system, but is chosen because of its simplicity. The nonlinear model is given by the following ordinary differential equation:

$$\frac{dx}{dt} = -x^2 + u. \tag{5.13}$$

25 batches are simulated using different starting values x_0 and different actuator trajectories. The data are sampled at $N = 41$ points during the batch. Examples of the inputs and outputs are shown in figures 5.1 and 5.2 for three batches. The full description of the example and the simulated data is shown in section 5.9.

In order to exite the system the input is varying, but known. It has been chosen that the input should be a linear interpolation between (random) values at $t = 0, 3, 5, 7, 10$ as illustrated in figure 5.1. These input signals may not be optimal for the identification, but are selected because these piecewise linearly varying signals are easy to implement on process control systems. The selection of the optimal input strategy and implementation of the signals on a real plant is left to further studies. The signals used for the identification are noise free.

The simple model that only requires past inputs (equation (5.6)) is estimated first. The resulting $\mathbf{B}$ matrix is visualised in figure 5.3 as a contour plot. The contour plot is used in this chapter to demonstrate the structure of the matrices. The structure does not depend on the number of samples obtained during the batch and the individual parameter values (i.e. contour levels) are not important for this demonstration. Each row corresponds to a time instant during the batch. It is seen that the value of the model parameters is large along and near the diagonal of $\mathbf{B}$. These elements correspond to the input at time t having the major effect on the outputs measured shortly *after* and shortly *before* time t. The three peaks that are seen along the diagonal are caused by the input signal that switches direction at these points in time. A simple remedy is to change the input signal such that changes are not introduced at the same time for all batches in the data set that is used for identification unless the model is developed especially to reflect such switches e.g. when it is known that the batch will always have input changes at specific times, which is a relevant assumption for batches that follow a recipe.

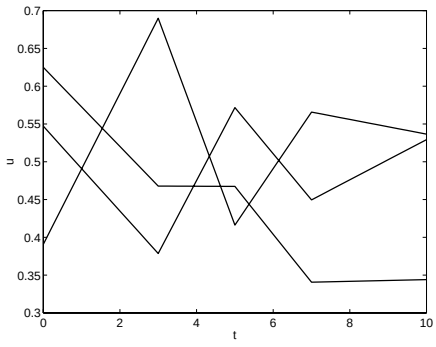


Figure 5.1. Input data for three batches used for the parameter estimation.

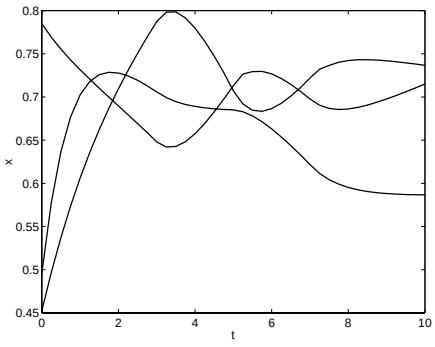


Figure 5.2. Raw output data for three batches used for the parameter estimation.

To validate the model a simulation is conducted with an actuator trajectory that is different from those used for model estimation; it has a peak at $t = 5$ with a higher maximum value than used for the model estimation (see figure 5.4). The input signal for the validation is similar to the input signal used for the estimation in the sense that is varies piecewise linearly, but the switching times are different. This input trajectory will be used also for the following SISO examples.

The result of the simulation can be seen in figure 5.4 and 5.5. When the initial value x_0 equals the mean of those used for model estimation as in figure 5.4 the prediction is quite good throughout the batch. However, one can in general not rely on being able to reproduce this desired initial condition for every batch even though it is attempted in production. The result so far is therefore optimistic. When the initial value is changed from the desired initial condition as shown in figure 5.5 the predictions of the system output are severely biased during the first half of the batch, but as soon as the effect of the biased initial conditions on the model estimate disappears the estimate becomes much more accurate.

The model $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}$ is estimated leading to the matrices $\mathbf{A}$ and $\mathbf{B}$ which are depicted in figures 5.6 and 5.7 as contour plots. Again it is seen that the matrices have their main non-zero content at and near the diagonal. Peaks in the $\mathbf{A}$ and $\mathbf{B}$ matrices are caused by the energy in the input signal used for the estimation. The non-zero elements seen in the simple model in the top rows of $\mathbf{B}$ have now disappeared since the effect of these non-zero elements are now included into the modelling of the effect of x_0 through $\mathbf{A}$.

For this more advanced model a simulation that is similar to the one performed for the simple model is carried out. The simulation is performed recursively in the following way. The known initial value is used for the construction of the vector $\mathbf{x}_{k-1}$. The remaining values are filled with the mean values from

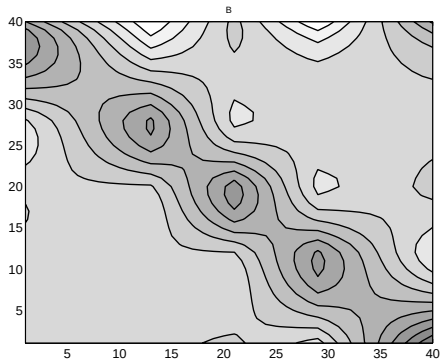


Figure 5.3. Contour plot for the $\mathbf{B}$ (40×40) matrix for the simple model $\mathbf{x}_k = \mathbf{B}\mathbf{u}_{k-1}$. Three peaks are seen. These peaks are caused by the input signal used for the estimation.

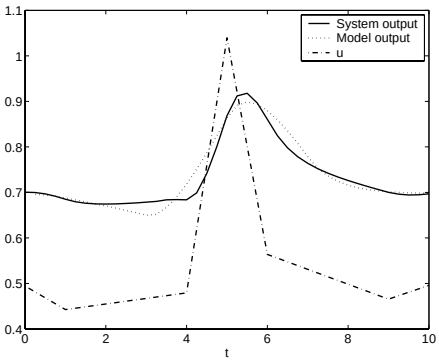


Figure 5.4. Simulation result for an unusual input trajectory using the simple model $\mathbf{x}_k = \mathbf{B}\mathbf{u}_{k-1}$. The initial state x_0 is for this simulation is equal to the mean used for model estimation. SSQ $= 3.66 \cdot 10^{-4}$.

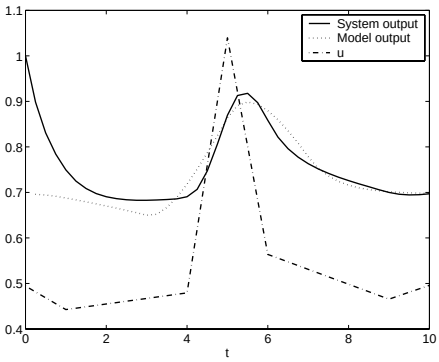


Figure 5.5. Simulation result for an unusual input trajectory using the simple model $\mathbf{x}_k = \mathbf{B}\mathbf{u}_{k-1}$. The initial state x_0 is 1. SSQ $= 0.0023$.

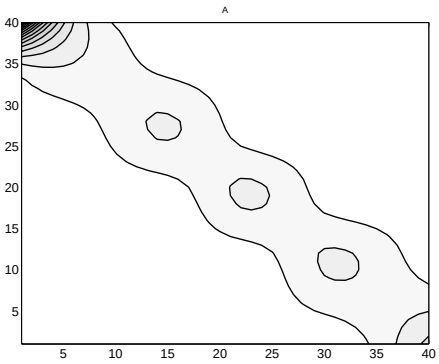


Figure 5.6. **A** for the model $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}$

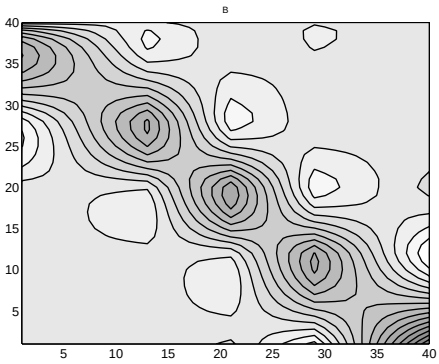


Figure 5.7. **B** for the model $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}$

the data that are used for the model estimation. Using the model equation we obtain a new vector $\mathbf{x}_k$. The values from this vector are used to refine the vector $\mathbf{x}_{k-1}$ since we now have a better guess for the output values than merely the mean values. This direct substitution in the right hand side of the model equation is performed recursively until the output vector have reached stationarity or if the model is not stable the recursion continues until there is negligible change in $\mathbf{x}_k$. The simulation is treated in greater detail in section 5.5.

The simulation results are shown in figures 5.8 and 5.9 for three recursions of the model equations. It is seen that the model output is very close to the system output for the major part of the duration of the batch. The model type with both **A** and **B** matrices is able to handle variation in the initial condition, which was not handled well by the model with only a **B** matrix.

5.4.2 Causality Constraints

By imposing structure on the matrices based on causality and other process knowledge the number of parameters to be estimated in the model matrices can be significantly reduced from the general case presented in the previous section.

For state space models of the type $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}$ some constraints can be imposed if a causal model is desired. The developed models so far all have the possibility to have non-zero elements above and below the diagonal in the **A** and **B** matrices. In order to ensure a causal model only non-zero elements can be permitted on and below the diagonal in both model matrices.

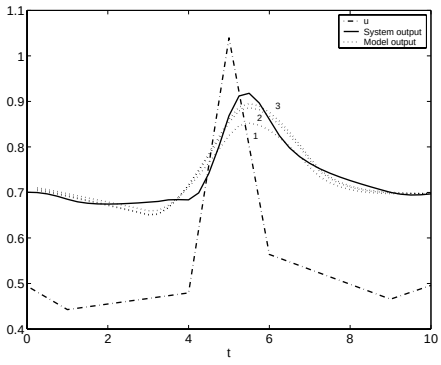


Figure 5.8. Simulation result for validation of the ARX model. The initial state x_0 is for this simulation equal to the mean of the ones used for model estimation. SSQ $= 3.48 \cdot 10^{-4}$. The figure shows 3 recursions of the model equations.

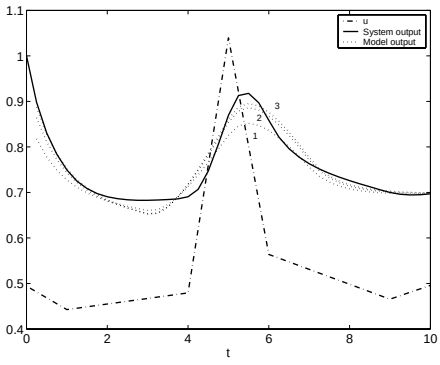


Figure 5.9. Simulation result for validation of the ARX model. The initial state x_0 is 1. SSQ $= 3.30 \cdot 10^{-4}$. The figure shows 3 recursions of the model equations.

Furthermore, based on physical knowledge for dissipative systems the values in the matrices should decrease as they get further away from the diagonal as these values indicate interaction between measurements that are far away in time.

To obtain a causal model, equation (5.7) is rewritten with explicit elements where the new causal structure of the model is introduced

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & \mathbf{0} \\ a_{21} & a_{22} & \vdots \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & \mathbf{0} \\ b_{21} & b_{22} & \vdots \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix}. \tag{5.14}$$

The challenge is to estimate the non-zero elements in the matrices $\mathbf{A}$ and $\mathbf{B}$ while maintaining the value zero where intended. The model (5.14) can be rewritten into

$$x_1 = a_{11}x_0 + b_{11}u_0 \tag{5.15}$$
$$x_2 = a_{21}x_0 + a_{22}x_1 + b_{21}u_0 + b_{22}u_1 \tag{5.16}$$
$$x_3 = a_{31}x_0 + a_{32}x_1 + a_{33}x_2 + b_{31}u_0 + b_{32}u_1 + b_{33}u_2, \tag{5.17}$$

where it is obvious that outputs are only affected by past outputs and inputs. These equations result in the following equation system where the parameters to be estimated are extracted as a vector

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_0 & & & & & & u_0 & & & & & \\ & x_1 & & x_0 & & & & u_1 & & u_0 & & \\ & & x_1 & & x_1 & x_0 & & & u_2 & & u_1 & u_0 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{22} \\ a_{33} \\ a_{21} \\ a_{32} \\ a_{33} \\ b_{11} \\ b_{22} \\ b_{33} \\ b_{21} \\ b_{32} \\ b_{33} \end{bmatrix} \tag{5.18}$$

$$\mathbf{x}_3 = \mathbf{F}\boldsymbol{\theta}, \tag{5.19}$$

where $\mathbf{F}$ contains the measured input and output data. Note that $\mathbf{F}$ in general will be sparse. The parameter estimation problem can be solved in various ways. Using the pseudo inverse based on the SVD is one of them. This leads to the solution for the general problem:

$$\boldsymbol{\theta} = \mathbf{F}^{\#}\mathbf{x}_k. \tag{5.20}$$

When this solution is found using the SVD decomposition of $\mathbf{F} = \mathbf{USV}^{\top}$ the matrices $\mathbf{U}$ and $\mathbf{V}$ generally become dense even when $\mathbf{F}$ is sparse. For a large

number of parameters this can become prohibitive. Instead special methods for sparse matrices are recommended. Such methods are available in the Regularisation Toolbox for MATLAB [Hansen, 1998].

The Regularisation Toolbox contains many routines that can be used to solve the posed estimation problem. One is based on the conjugate gradient algorithm and is called CGLS. The CGLS algorithm has two very good qualities:

- It solves the least squares problem without changing $\mathbf{F}$.
- When the iterations are stopped short of convergence a *regularised* solution is obtained [Hansen, 1996]. It finds the solution with the initial guess $\boldsymbol{\theta} = \mathbf{0}$ (which corresponds to a very regularised solution) and during the iterations it converges towards the least squares solution.

To continue the SISO example with the causality constraints results are here given for an estimation problem both using the pseudo inverse based on the SVD and using CGLS. First the singular values of $\mathbf{F}$ are found and these are plotted in figure 5.10.

It is seen that there is a sharp drop in the singular values at the 171st component and this value is used for finding the pseudo inverse of $\mathbf{F}$, which again is

Algorithm 1: CGLS algorithm. Solution of the (sparse) least squares problem $\|\mathbf{Ax} - \mathbf{b}\|$ using the conjugate algorithm.

(1) $\mathbf{x}^{(0)} = \mathbf{0}$, $\mathbf{d}^{(0)} = \mathbf{A}^\top\mathbf{b}$, $\mathbf{r}^{(0)} = \mathbf{b}$
(2) $\alpha_k = \|\mathbf{A}^\top\mathbf{r}^{(k-1)}\|_2^2 / \|\mathbf{A}\mathbf{d}^{(k-1)}\|_2^2$
(3) $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \alpha_k\mathbf{d}^{(k-1)}$
(4) $\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - \alpha_k\mathbf{A}\mathbf{d}^{(k-1)}$
(5) $\beta_k = \|\mathbf{A}^\top\mathbf{r}^{(k)}\|_2^2 / \|\mathbf{A}^\top\mathbf{r}^{(k-1)}\|_2^2$
(6) $\mathbf{d}^{(k)} = \mathbf{r}^{(k-1)} + \beta_k\mathbf{d}^{(k-1)}$
(7) **if** $\mathbf{x}$ has not converged
(8) **goto** 2.

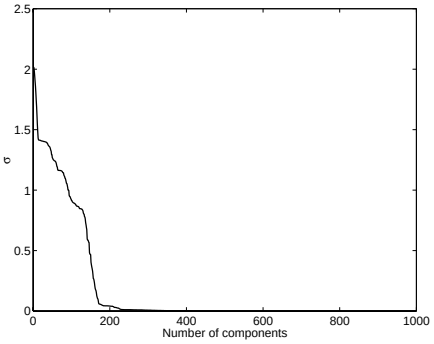


Figure 5.10. Singular values of $\mathbf{F}$ for the SISO example using a causal model.

used for finding **A** and **B**. The condition number of **F** is $2.1 \cdot 10^{19}$ which rules out using the ordinary least squares for this problem. The calculation of the pseudo inverse of **F** takes a significant amount of time. Although **F** is sparse the resulting matrices of the SVD are not sparse and become very large since 1638 parameters have to be estimated for this problem. For real life problems the matrices are expected to become much larger. This is a severe hindrance to using the SVD for these estimation problems.

The model matrices estimated using the SVD are shown in figures 5.11 and 5.12 as contour plots. It is seen that these matrices now only have non-zero content at and below the diagonal and that, especially for the **A** matrix, the values become very small when they are far from the diagonal.

Another way of displaying the content of the matrices is to plot the diagonals and subdiagonals in a simple plot. This is illustrated in figure 5.13. The effect of the input signal is seen very clearly in both **A** and **B**. In this plot it is possible to see the actual values of the most important parameters that are large and close to the diagonal. The sum of squares (SSQ) is for this model $2.05 \cdot 10^{-4}$.

The CGLS method involves iterations so the first thing is to find the optimal number of iterations, which is determined by cross validation. 200 iterations are performed and for each iteration the resulting model is compared to the validation simulation using the SSQ. A plot of the SSQ is shown in figure 5.14. Here it is seen that the minimum value is obtained at about 11 iterations, but the SSQ-curve is flat so the actual number can be varied without much change in the model prediction capability. The increase that is seen in the SSQ-curve is due to the solution becoming too close to the least squares solution and that there thus is a loss of the regularisation effect of the CGLS method.

The model matrices are shown in figure 5.15 and 5.16. Again the diagonal and subdiagonal elements have the largest magnitude. This leads to a plot of

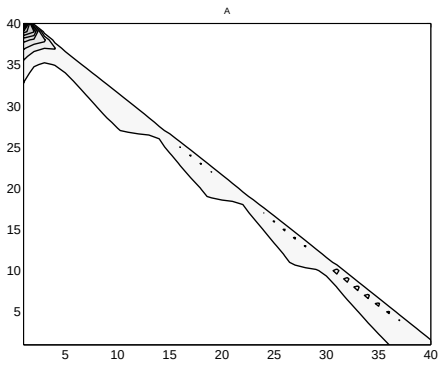


Figure 5.11. **A** matrix for the causal stacked state space model using the SVD.

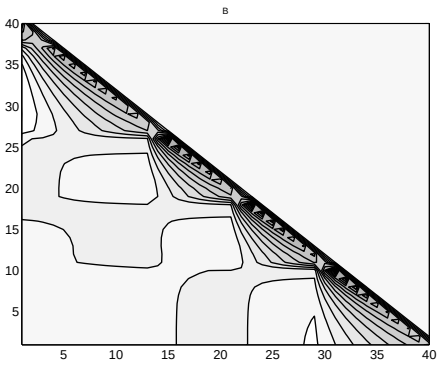


Figure 5.12. **B** matrix for the causal stacked state space model using the SVD.

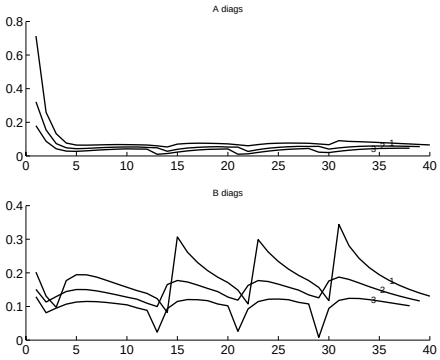


Figure 5.13. Plot of the diagonal and first three subdiagonals of **A** and **B** model matrices found using the SVD.

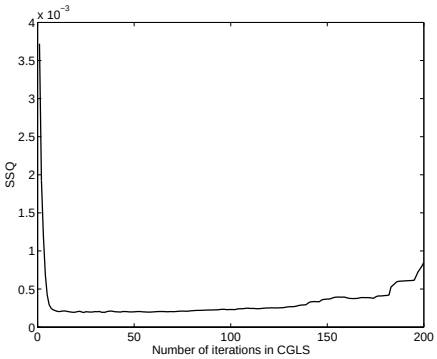


Figure 5.14. Sum of squares curve for the CGLS method.

the diagonals and first few subdiagonals only as demonstrated in figure 5.17. By comparison with figure 5.13 that shows the diagonals obtained using the SVD it is seen that the effect of the input signal still appears in the model, but the effect is much smaller. It has thus been demonstrated that the CGLS has a regularising effect. The CGLS requires in this case only 11 iterations and is much faster than using the SVD for this problem. The accuracy is the same as the SSQ obtained for the validation example is $2.04 \cdot 10^{-4}$.

5.4.3 Model Order

For simple ARX models the notion of model order describes the number of past sampled values that should be included in the model. The model order is described by the values n_a and n_b introduced in equation (5.1). The optimal

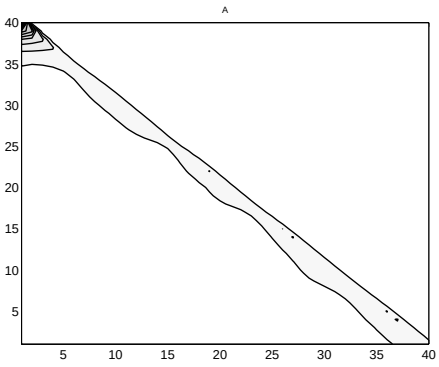


Figure 5.15. **A** matrix for the causal stacked state space model using CGLS.

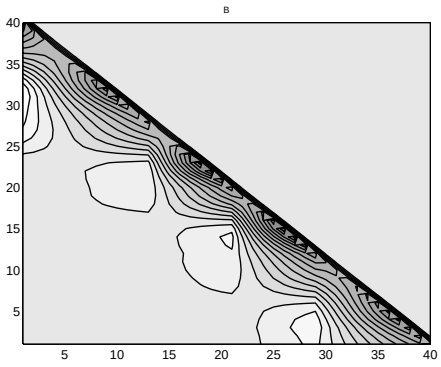


Figure 5.16. **B** matrix for the causal stacked state space model using CGLS.

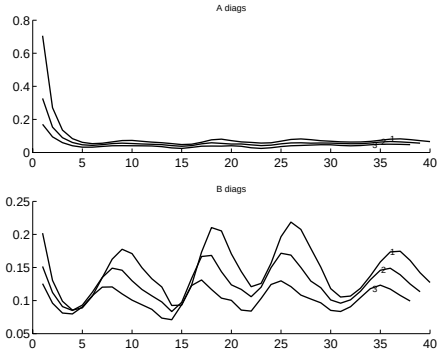


Figure 5.17. Plot of the dianogal and subdiagonals for the model matrices **A** and **B** obtained from CGLS

model order is determined by the sample time, the dynamics of the system, the frequency spectrum of the input signal etc. In reality the optimal model order will also be affected by the signal to noise ratio of the data and the number of samples and batches available.

By keeping the order of the ARX model low it is assured that the variance of the identified parameters is kept at a minimum. Such constraints can increase robustness of the identified model. One must on the other hand not unnecessarily restrict the order of the model since that will limit the ability of the model to follow the dynamic behaviour of the system.

For the stacked state space models the order of the ARX model can be transformed into having a model where only the diagonal and a few sub-diagonals are non-zero. For each row the number of parameters is the local order of the stacked state space model. This number does not have to be constant throughout the batch, but can change depending on the local dynamic behaviour of the batch in certain time intervals.

An rough initial estimate of the model order may be determined from knowledge of the process dynamics and the sample time. By estimating a model with full order an improved estimate of the model order can be obtained by looking at the magnitude of the parameters in the model with full order. It has been the experience in this work that the model order can be selected *much lower* than the contour plots of the full models indicate. Probably because the models with reduced order are morphysical realistic than models with full order. The selected model order can finally be validated using cross validation.

The SISO example from the previous sections is repeated for a model with relatively low model order. The model is simply selected to be $n_a = 4$ and $n_b = 4$ with little analysis of the suitability of this selection. The number of parameters become much smaller than in the previous examples and it is expected that the model improves as the information in the data is used for estimating a more parsimonious model.

The identification method using the SVD is presented first. The plot of the singular values in figure 5.18 shows that the optimal number of components is either 49 or 91 depending on which knee in the curve is chosen as being significant. Two simulations of the validation example for the two cases reveal that the SSQ becomes $5.49 \cdot 10^{-4}$ and $1.23 \cdot 10^{-4}$ for 49 and 91 components, respectively.

It is of course also possible to carry out the model estimation for the low order model using CGLS. First the optimal number of iterations must be found. Using the curve in figure 5.20 of the SSQ for a set of iterations using the validation example it is seen that the optimal number of iterations is about 50. This number of iterations lead to an SSQ equal to $6.16 \cdot 10^{-5}$ which is the lowest SSQ obtained so far for this system.

The diagonals of the model matrices are shown in figure 5.21. For the obtained low order model it is now possible to plot the parameters of the entire model in one single plot. The number of parameters have thus been signifi-

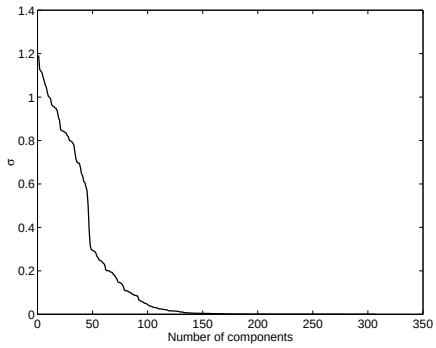


Figure 5.18. Plot of the singular values for a causal low order stacked state space model. The order of the model is $n_a = 4$ and $n_b = 4$.

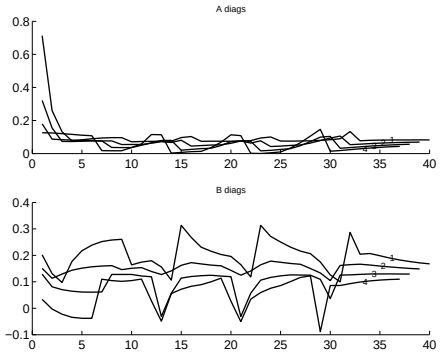


Figure 5.19. Diagonal plot of the model matrices for a causal low order stacked state space model found using the SVD. The order of the model is $n_a = 4$ and $n_b = 4$.

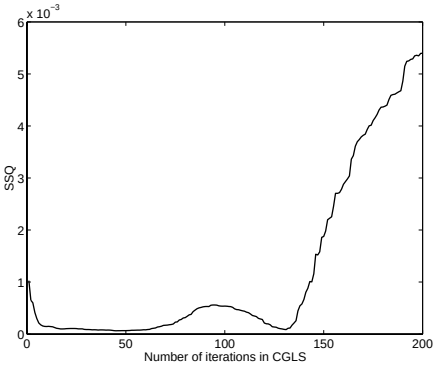


Figure 5.20. Plot of the SSQ for a causal low order stacked state space model found using CGLS. The order of the model is $n_a = 4$ and $n_b = 4$.

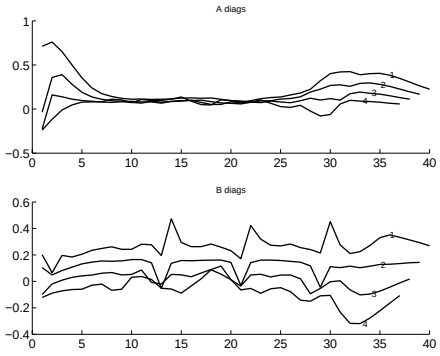


Figure 5.21. Diagonal plot of the model matrices for a causal low order stacked state space model found using CGLS. The order of the model is $n_a = 4$ and $n_b = 4$.

cantly reduced compared to the full non-causal models and the performance has been improved significantly too.

5.4.4 ARX with constant parameters

If the system is stationary after being linearised around the normal operating trajectory and the dynamics of the system is linear and time invariant it is possible to use a standard ARX implementation for estimating the parameters

$$\begin{aligned} y(n) = a_1 y(n-1) + a_2 y(n-2) + \cdots + a_{n_a} y(n-n_a) \\ + b_1 u(n-1) + b_2 u(n-2) + \cdots + b_{n_b} u(n-n_b). \end{aligned} \quad (5.21)$$

When the parameters in this model type are identified using PCA it is called Dynamic PCA [Wise, 1991; Simoglou *et al.*, 2002; Juricek *et al.*, 1999; Luo *et al.*, 1999]. When PLS is used it is called dynamic PLS [Simoglou *et al.*, 1999; Lakshminarayanan *et al.*, 1997].

When a batch system is modelled with this model type special care must be introduced at the beginning of the batch since the necessary n_a or n_b (whichever is the largest) samples are not available. There may even be extra time delays between the input and output that would further set back the start of the model when the left hand side $y(n)$ can be estimated. A solution is obtained by writing an equation system similar to the system given in equation (5.14) that takes the starting problem into consideration.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_1 & & \mathbf{0} \\ a_2 & a_1 & \\ a_3 & a_2 & a_1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 & & \mathbf{0} \\ b_2 & b_1 & \\ b_3 & b_2 & b_1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix}. \quad (5.22)$$

This model can be rewritten into

$$x_1 = a_1 x_0 + b_1 u_0 \quad (5.23)$$

$$x_2 = a_2 x_0 + a_1 x_1 + b_2 u_0 + b_1 u_1 \quad (5.24)$$

$$x_3 = a_3 x_0 + a_2 x_1 + a_1 x_2 + b_3 u_0 + b_2 u_1 + b_1 u_2, \quad (5.25)$$

which finally leads to the simple linear equation

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_0 & & & u_0 & & \\ x_1 & x_0 & & u_1 & u_0 & \\ x_2 & x_1 & x_0 & u_2 & u_1 & u_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (5.26)$$

$$\mathbf{x}_3 = \mathbf{F}\boldsymbol{\theta} \quad (5.27)$$

This model formulation has fewer parameters than the stacked state space model stated in equation (5.18). This is even more evident as the model size increases.

5.4.5 Regularisation

The above mentioned approach of simplifying the model can restrict the dynamic behaviour of the model and introduce bias. Restricting the model by eliminating parameters may be reasonable when either too little data or no prior information is available. Instead of eliminating information it is usually a better strategy to integrate available information into the modelling methodology.

In the estimation of dynamic models there can be several reasons why the problem is ill-conditioned:

- Too small excitation of the input signal.
- The variables of the input or output signal become too correlated. Furthermore, correlation between time shifted variables can occur. This is especially a problem when the model order is too high.
- When there are too many parameters.
- When there are too little data. Either too few samples taken during a batch or data from too few batches.

The problem is said to be ill-conditioned when the solution of the model equations is not unique. Ill-conditioning leads to large parameter variance, which has the effect that it is only possible to determine some parameters accurately. For correlated parameters it may only be possible to determine the sum and not their distinct values. In the end a model may result that is not physically realistic.

One way to overcome the problem of ill-conditioning is to use *regularisation* [Hansen, 1996]. Regularisation is a way to deal with the bias/variance dilemma. This dilemma explain the trade-off between bias and variance that exists when estimating parameters. For many applications it can be justified to introduce a little bias into the model in order to reduce the variance of the parameters if this change of the estimation procedures reduces the overall mean square error (see appendix B.4).

5.4.5.1 Smoothing of the parameters

All available prior information about the behaviour of the system should be included in the solution strategy if possible. However, it is often difficult to represent the prior knowledge in a format that is compatible with the ARX models described.

One way of introducing prior information is to use regularisation [Hansen, 1996]. Regularisation introduces extra information directly in the solution equations by adding extra terms. The most used method for regularisation is Tikhonov regularisation. The unregularised solution to the parameters estimation problem in the model $\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_{t-1}$ is given by the equation system $\mathbf{x}_t = \mathbf{F}\boldsymbol{\theta}$ corresponding to the minimisation

$$\min_{\boldsymbol{\theta}} \|\mathbf{F}\boldsymbol{\theta} - \mathbf{x}_t\|^2 . \tag{5.28}$$

The Tikhonov regularised solution is given by

$$\min_{\boldsymbol{\theta}} \left\{ \|\mathbf{F}\boldsymbol{\theta} - \mathbf{x}_t\|^2 + \lambda^2 \|\mathbf{L}\boldsymbol{\theta}\|^2 \right\}. \tag{5.29}$$

$\mathbf{L}$ is often chosen to be the identity matrix or a more general diagonal matrix and the regularisation has in this case the effect of decreasing the absolute value of the parameters. Other choices of $\mathbf{L}$ are possible. It is common to make $\mathbf{L}$ a discrete approximation of a derivative operator and the regularisation will then have the effect of a low pass filter, e.g.:

$$\mathbf{L}_1 = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix} \tag{5.30}$$

$$\mathbf{L}_2 = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{bmatrix} \tag{5.31}$$

Observing the content of $\boldsymbol{\theta}$ in equation (5.18) it can be seen that $\boldsymbol{\theta}$ contains blocks of diagonals. This means that the $\mathbf{L}$ matrix cannot have the structure indicated in (5.30) and (5.31), but rather contains blocks of these matrices along the diagonal. This matrix is termed $\boldsymbol{\Lambda}$:

$$\boldsymbol{\Lambda} = \begin{bmatrix} \mathbf{L}_{1(n)} & & & & & & & & \\ & \mathbf{L}_{1(n-1)} & & & & & & & \\ & & \ddots & & & & & & \\ & & & \mathbf{L}_{1(n-n_a)} & & & & & \\ & & & & \mathbf{L}_{1(n)} & & & & \\ & & & & & \mathbf{L}_{1(n-1)} & & & \\ & & & & & & \ddots & & \\ & & & & & & & \mathbf{L}_{1(n-n_b)} \end{bmatrix}, \tag{5.32}$$

where the notation $\mathbf{L}_1(n)$ means a discrete approximation to a first derivative operator with n columns.

5.5 Simulation

The models that have been developed sofar is a dynamic model of the batch process. This model type can be used for monitoring, control and optimisation of the process. A prerequisite of using the model for these tasks is the ability to use the model for simulation.

5.5.1 Motivation

One of the many uses of the developed models is to be able to simulate the system. It is a minimum requirement for the models that they are able to simulate the system for input values and initial values (x_0) that are close to the ones used when estimating the model.

In the following section a direct substitution method for simulation is described. Subsequently an analytical solution is developed that is extended into handling state constraints.

Using the model equation it is easy to see how the model can be used for simulation when x_0 and $\mathbf{u}_{k-1}$ are known. The model equations

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \tag{5.33}$$

are simulated by inserting the known x_0 in $\mathbf{x}_{k-1}$ having zeros in the remaining places since $\mathbf{x}$ is in deviation variables. $\mathbf{x}_k$ can then be calculated. The calculations are performed recursively by direct substitution until convergence or until the change in $\mathbf{x}_k$ is negligible.

Simple stability theory for LTI systems is used to assess the convergence properties of the iterations although the iterations have nothing to do with conventional simulation since k does not denote time in the calculations, but is simply viewed as an iteration index. A system $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k$ is stable if and only if all the eigenvalues of $\mathbf{A}$ fall inside the unit circle [Rugh, 1996].

5.5.2 Closed form

The recursive solution method can be avoided by rewriting the system into a form that leads to a closed form solution.

The ARX model is written

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}. \tag{5.34}$$

x_0 and $\mathbf{u}_{k-1}$ are assumed known. This is introduced into equation (5.34)

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{c}, \tag{5.35}$$

where $\mathbf{c} = \mathbf{B}\mathbf{u}_{k-1}$. The matrices and vectors are expanded to show their elements for a small example with $n = 4$ and $n_a = 2$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ 0 & a_{32} & a_{33} & \\ 0 & 0 & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}, \tag{5.36}$$

which is simplified into

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} a_{11}x_0 \\ a_{21}x_0 + a_{22}x_1 \\ a_{32}x_1 + a_{33}x_2 \\ a_{43}x_2 + a_{44}x_3 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}. \tag{5.37}$$

It is now the goal to extract an $\mathbf{x}$ vector on the left hand side that is the same as the one found on the right hand side. This is possible by moving the terms involving x_0 to the $\mathbf{c}$ vector.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ a_{22} & \ddots & \ddots & \vdots \\ a_{32} & a_{33} & \ddots & \vdots \\ \mathbf{0} & a_{43} & a_{44} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} c_1 + a_{11}x_0 \\ c_2 + a_{21}x_0 \\ c_3 \\ c_4 \end{bmatrix} \tag{5.38}$$

$$\mathbf{x} = \tilde{\mathbf{A}}\mathbf{x} + \tilde{\mathbf{c}} \tag{5.39}$$

The following equation system is then formed, which can be used for the calculation of $\mathbf{x}$

$$(\mathbf{I} - \tilde{\mathbf{A}})\mathbf{x} = \tilde{\mathbf{c}}. \tag{5.40}$$

Note that the vector $\tilde{\mathbf{c}}$ depends only on initial state x_0 and the input $\mathbf{u}$. This means that we now have a very simple linear model for simulating a batch process. The matrix on the left hand side

$$\tilde{\mathbf{L}} = (\mathbf{I} - \tilde{\mathbf{A}}) \tag{5.41}$$

is a lower triangular matrix and the equation system can be solved using forward substitution.

For analysis purposes it is also convenient to give the explicit result for $\mathbf{x}$ as

$$\begin{aligned} \mathbf{x} &= \tilde{\mathbf{L}}^{-1}(\mathbf{A}_0 x_0 + \mathbf{B}\mathbf{u}) \\ &= \tilde{\mathbf{L}}^{-1} \begin{bmatrix} \mathbf{A}_0 & \mathbf{B} \end{bmatrix} \begin{bmatrix} x_0 \\ \mathbf{u} \end{bmatrix} \\ &= \mathbf{G} \begin{bmatrix} x_0 \\ \mathbf{u} \end{bmatrix} \\ &= \mathbf{G}\tilde{\mathbf{u}} \end{aligned} \tag{5.42}$$

where $\mathbf{G}$ is the matrix operator that maps known initial conditions and inputs contained in $\tilde{\mathbf{u}}$ to the outputs.

In the case of very uncertain models one may obtain a model that lead to physically impossible output estimates. By introducing constraints on the outputs a realistic output estimate can be obtained. Let $\mathbf{x}^*$ denote the unconstrained solution to the output estimation (simulation) problem. Then a constrained solution may be found using the *quadratic program* (QP)

$$\min_{\mathbf{x}} (\mathbf{x} - \mathbf{x}^*)^\top (\mathbf{x} - \mathbf{x}^*) \tag{5.43}$$

$$\text{s.t.} \quad x_{min,i} \leq x_i \leq x_{max,i} \tag{5.44}$$

Commercial algorithms for performing QP are available that will solve this problem in very little time compared to the sample time of biochemical processes. This leaves sufficient time to solve the problem for each sample during the batch although the calculation times naturally increases as the size of the model increases.

5.6 MIMO models for batch

In the multivariate case the derivation of the equations become slightly more involved than in the univariate case, but is essentially the same procedure once the structure of the data matrices have been defined.

The primary type of unfolding used for the stacked state space models is to unfold the data such that the unfolded data matrix contains the measurements as separate blocks. Other choices of unfolding could have been made. The choice is made to ease the step from SISO models to MIMO models and to facilitate the construction of the **L** matrices used for regularisation introduced in section 5.4.5. The structure of the resulting matrices involved in the estimation is shown in the next section.

5.6.1 Matrix Structure

The structure of the **A** and **B** matrices is derived by observing that the MIMO data matrix is simply a set of measurements at each sample instant stacked on top of each other. The **A** and **B** matrices are then simply stacked as well. An example of the structure of the matrices is shown in figure 5.23 for a system with 3 outputs, 1 input and only 3 samples obtained during the batch. The system order is 2 for each subblock in the **A** and **B** matrices. The order doesn't have to be equal for the wubbloxks and may vary with time, but this issue is not persued further here. The number of elements in **A** is $n_s^2K^2$, where n_s is the number of measured output variables, but the matrix is sparse and contains only $n_s^2 n_a(2K - n_a + 1)/2$ non-zero elements. Eliminating the K^2 term in the storage of the matrix can assist greatly when many samples are obtained from a batch since K will usually be much larger than n_s, n_a and n_b. It does not

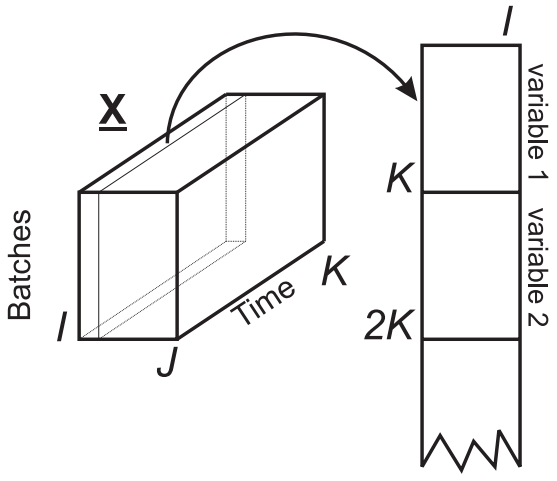


Figure 5.22. Unfolding into a structure for MIMO stacked state space modelling.

seem possible to eliminate the term n_s^2 unless one observes a loosely connected system where many subblocks in the $\mathbf{A}$ matrix are known to be exactly equal to zero. $\mathbf{B}$ contains $n_s n_u n_b(2K - n_b + 1)/2$ non-zero elements.

The generation of the $\mathbf{F}$ (see equation (5.19)) and $\tilde{\mathbf{L}}$ (see equation (5.41)) matrices used for the regularised estimation procedure is straightforward. It is simply a matter of stacking the matrices on top of each other and keeping an eye on the index of the matrix elements.

5.6.2 Example

A fermentation process in fed-batch operation is here used as an axample of the MIMO modelling using a simple unstructured biomass model as reference model.

$$\dot{S} = -\frac{\mu}{0.5}X + (S_f - S)\frac{F}{V} \tag{5.45}$$

$$\dot{X} = \mu X - X\frac{F}{V} \tag{5.46}$$

$$\dot{V} = F, \tag{5.47}$$

where $\mu = \frac{\mu_{max}S}{K+S+0.5S^2}$. The example is described in greater detail in section 5.10

This model is simulated for 4 hours. The reference initial conditions x_0 are $\begin{bmatrix} 0.2 & 1 & 1 \end{bmatrix}$, but when creating the data sets the initial values will be varied using the specified values as mean values and a standard deviation of $\begin{bmatrix} 0.025 & 0.1 & 0.1 \end{bmatrix}$. All the data are given in section 5.10. In this section only the data from the three first batches are shown for clarity. Figure 5.24 shows the input signal for 3 batches and figure 5.25 shows the corresponding output for the same batches. The sample time is 0.1 hours leading to 41 samples for the entire batch. It is seen that the input signal for the MIMO estimation is created like

$$\mathbf{A} = \left[\begin{array}{ccc:ccc:ccc}
a_1 & 0 & 0 & a_6 & 0 & 0 & a_{11} & 0 & 0 \\
a_4 & a_2 & 0 & a_9 & a_7 & 0 & a_{14} & a_{12} & 0 \\
0 & a_5 & a_3 & 0 & a_{10} & a_8 & 0 & a_{15} & a_{13} \\
\hdashline
a_{16} & 0 & 0 & a_{21} & 0 & 0 & a_{26} & 0 & 0 \\
a_{19} & a_{17} & 0 & a_{24} & a_{22} & 0 & a_{29} & a_{27} & 0 \\
0 & a_{20} & a_{18} & 0 & a_{25} & a_{23} & 0 & a_{30} & a_{28} \\
\hdashline
a_{31} & 0 & 0 & a_{36} & 0 & 0 & a_{41} & 0 & 0 \\
a_{34} & a_{32} & 0 & a_{39} & a_{37} & 0 & a_{44} & a_{42} & 0 \\
0 & a_{35} & a_{33} & 0 & a_{40} & a_{38} & 0 & a_{45} & a_{43}
\end{array}\right] \quad \mathbf{B} = \left[\begin{array}{ccc}
b_1 & 0 & 0 \\
b_4 & b_2 & 0 \\
0 & b_5 & b_3 \\
\hdashline
b_6 & 0 & 0 \\
b_9 & b_7 & 0 \\
0 & b_{10} & b_8 \\
\hdashline
b_{11} & 0 & 0 \\
b_{14} & b_{12} & 0 \\
0 & b_{15} & b_{13}
\end{array}\right]$$

Figure 5.23. $\mathbf{A}$ and $\mathbf{B}$ system matrices for a MIMO stacked state space model with 3 outputs, 1 input and 3 samples obtained during the batch. The order is 2 for both each block of the $\mathbf{A}$ and $\mathbf{B}$ matrices. The order does generally not have to be the same for all outputs or all inputs. The matrices are sparse and as the matrices become larger it becomes increasingly important to use a sparse storage scheme.

the ones used for the SISO model with the difference that switches between the piecewise linear signals are now performed at different (random) points in time during the batch.

It is assumed that the substrate and biomass concentrations and the volume can be measured at each sample time. In many industrial fermentation processes this is *not* possible, but the example is chosen because it is simple and yet poses some interesting problems concerning the described modelling methodology.

In the following analysis it is assumed that very little is known about the system in order to mimic the real situation where the system is complex and it is substantially more complicated to use prior knowledge about the measured quantities e.g. in the case where also pH, stirrer power, and temperature etc. are measured. On the other hand it is easy to see that there is no influence from the substrate or biomass concentrations to the volume. These types of interactions (or lack thereof) are in many cases easy to identify and may significantly reduce the number of parameters when such parameters are explicitly set to zero in the model instead of relying on the data material to provide that information. These assumptions must of course be verified as an unwarranted elimination of an interaction term may severely handicap the model performance. Section 5.10 gives contour plots and diagonal plots to justify the elimination of the interaction from S and X to V. Figure 5.41 illustrates that a low order model has very little interaction corresponding to setting the two lower left hand blocks in the $\mathbf{A}$ matrix to zero blocks, compare with figure 5.23.

The model is validated using a new data set obtained with a special input trajectory. See figure 5.26 and compare with figure 5.24. This unusual profile is used to validate the model.

The model is estimated using CGLS. The method using SVD will generally be too time consuming for the MIMO models. The optimal number of iterations

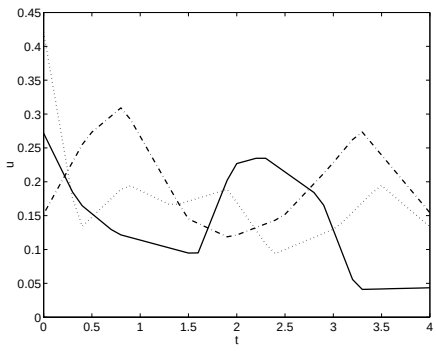


Figure 5.24. Input signal for 3 batches used for MIMO stacked state space model estimation.

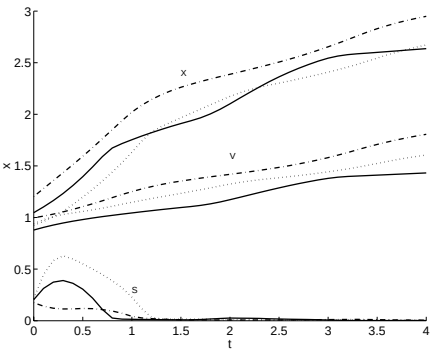


Figure 5.25. Output signal for 3 batches used for MIMO stacked state space model estimation.

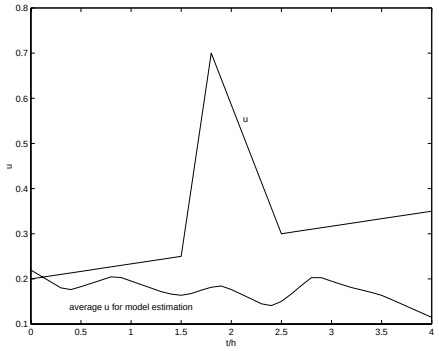


Figure 5.26. New input trajectory to validate the estimated model.

of the CGLS method is determined using figure 5.27 that shows the SSQ as a function of the number of iterations for the validation input. The curve is not as smooth as for the SISO case, but a minimum can be found at 240 iterations that give a SSQ at 0.0016. The simulated output is compared with the system output in figure 5.28. This figure shows that the model is able to capture the system behaviour well throughout the batch. It is expected that the accuracy is highest for the volume and biomass measurements and lower for the substrate measurements due to the faster dynamics af the latter.

The model parameter matrices are visualised in figures 5.29 and 5.30. In these two figures the diagonals of the **A** and **B** matrices are plotted for each subblock

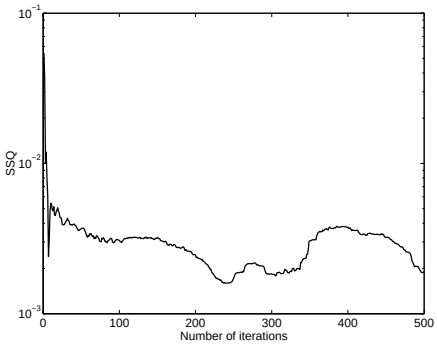


Figure 5.27. Logarithmic plot for finding the optimal number of iterations of the CGLS algorithm.

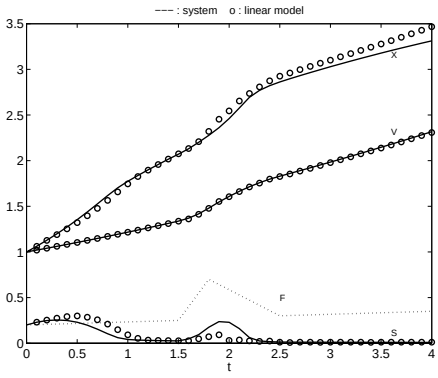


Figure 5.28. Simulation of the stacked state space model found using CGLS ($n_a = 3,\ n_b = 2$) for the validation example using the a constrained simulation method to maintain S positive.

of the matrices corresponding to the natural division of the matrices by the number of input and output variables. Figure 5.23 illustrates this division for a small example model.

Remember that the two lower left hand subblocks of **A** were set to zero hence elements in these subblocks are not estimated. Figure 5.29 show that the parameters have the largest variation in the beginning of the batch. This is the time where the process has the fastest dynamics due to the small volume at this time. After about one third of the batch most model parameters seem to have settled to a constant value. It is noticed that the model parameters for the volume in the lower right hand corner are almost constant throughout the entire batch except for some effects from the initial conditions. The sum of these parameters is one as expected since the volume only depends on F. Figure 5.30 that shows the diagonals of the **B** matrix also indicates that the model parameters change most at the beginning of the batch. As the volume increases the effect of the flow rate decreases.

5.7 Discussion

Linear models have been studied throughout the entire history of control theory [Bennett, 1997]. In the beginning this was because computer technology was not available and linear models and methods for handling linear models were the only practical way to handle the problems posed. With the computer power that is available today one would think that handling nonlinear models and nonlinear programming would not pose a problem. Unfortunately, nonlinear programming may give problems in terms of convergence and long and unpredictable computation times and complicated behaviours.

Nonlinear programming also requires much better models in order to give accurate results. This accuracy requirement is a problem when the models or model parameters are identified using data since the set of nonlinear models is larger than the set of linear models.

By using linear models most of the problems with the nonlinear models disappear without sacrificing modelling potential when operating in a region where the process can be considered linear, which may be the case when a conservative design of the operation of an industrial batch process is considered. Future designs where the process may follow constraints more closely and the process design allow larger deviations from the reference trajectory lead to more nonlinear systems which in turn may lead to a higher relevance and profitability of nonlinear (first principles) modelling.

The linear batch modelling principle investigated in this chapter is based on linear time invariant (LTI) models. This allows exploitation of the massive amount of theory for model analysis and design of control and optimisation algorithms.

Reducing the parameters estimation to a linear optimisation problem results in computational problems that are easily solved using rather simple and fast

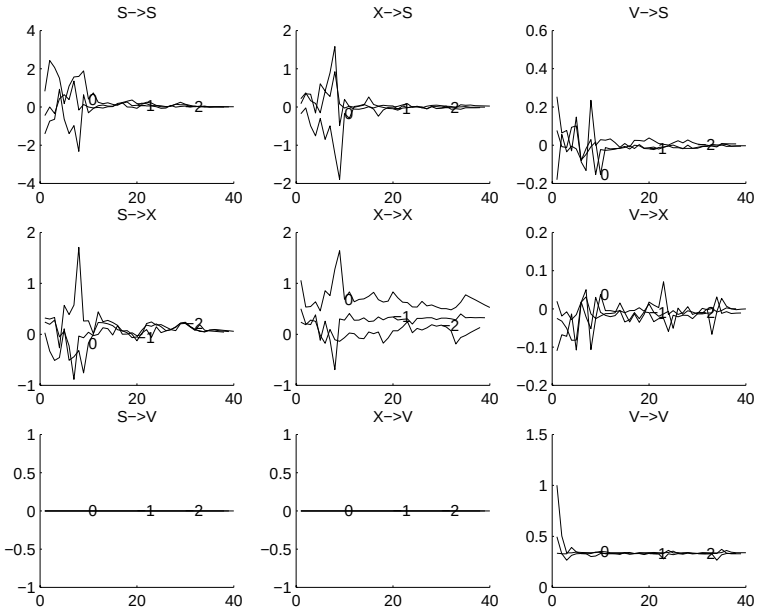


Figure 5.29. Plot of the diagonal and subdiagonals of the blocks in the estimated **A** matrix for the MIMO stacked state space model using CGLS. The order of the model is: $n_a = 3$ and $n_b = 2$.

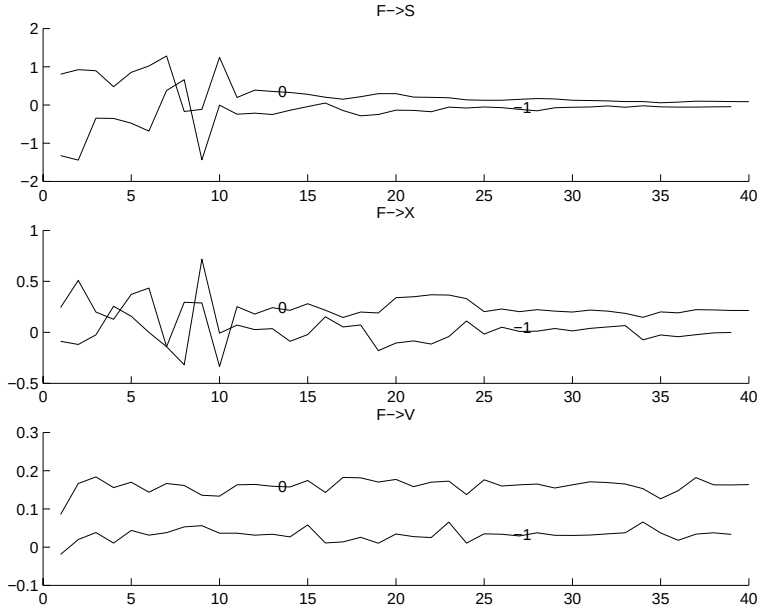


Figure 5.30. Plot of the diagonal and subdiagonals of the blocks in the estimated **B** matrix for the MIMO stacked state space model using CGLS. The order of the model is: $n_a = 3$ and $n_b = 2$.

algorithms. The solution can therefore be obtained in a very short time and what is perhaps even more important the calculation time is bounded such that control actions can be implemented in real-time. However, the number of parameters can be very large if the number of samples per batch is large and the number of measurements is also large. Only by utilising regularisation and choosing a proper model order (or model orders) can the parameter estimation problem be carried out with confidence. By including physical knowledge, especially when it is known that two variables are not affecting each other, the parameter estimation problem can be further reduced.

5.8 Conclusion

The main results of this chapter are:

- Development of a simple dynamic linear model structure tailored to batch processes based on an ARX structure. This model type is termed a stacked state space model, which is short for finite-horizon, time-shifted, stacked, linear state space models.
- Causality and other types of physical knowledge can explicitly be incorporated into the model structure.
- Estimation algorithms utilising regularisation can be used to increase the accuracy since it makes it possible to include some system knowledge and can be used to balance bias and variance in the model.

The developed models may directly be used for batch optimisation, optimising control and model predictive control [Lee *et al.*, 1997; Chin *et al.*, 1998].

5.8.1 Future Research

Research has to be done on the application of the identified model. The above mentioned areas of control and optimisation are natural choices for areas for the developed model type since it is a prediction model. The model may also be used for monitoring purposes. This work in currently in development. Initial findings will be described in the next chapter

For the modelling step it can be foreseen that some improvement can be made. The number of parameters in the current implementation is large, which may lead to a model with too large uncertainty in the parameters. One way of reducing this uncertainty is to use subspace methods [Liu, 1997; Verhaegen and Yu, 1995]. Since it is agued that the parameters in the estimated matrices are expected to vary slowly along the diagonals and subdiagonals it might also be possible to model the development along the diagonals and subdiagonals of the $\mathbf{A}$ and $\mathbf{B}$ matrices as splines (or other curves) once a suitable set of node points has been identified. The use of splines will greatly reduce the number of parameters. The regularisation has almost the same effect as using splines, though. Using regularisation with a fourth order derivative operator

approximation $\mathbf{L}_4$ with a very large value of the regularisation parameter λ will provide the effect of having a cubic spline. As it has been shown there can be a need for regularisation also using lower order derivative operator approximations. Functional data analysis is a recently developed statistical methodology that model data that is the output of a function using knowledge about the relationship of the data to specify parts of the model (e.g. expressions for the kinetics) [Ramsay and Silverman, 1997].

The linear models are used for modelling a nonlinear system. As long as the behaviour of the system is linear, which is the case in an operating region near the normal operating trajectory where the model has been identified the linearity assumption does not pose a problem. When there are large deviations and especially when nonlinearities are encountered such as limits in growth rates or operation near optimal conditions special handling is necessary because of the sign changes involved. The inclusion of nonlinearities in the modelling and/or control must be carried out in further research.

In order to handle the nonlinearities in the system switching schemes may be deviced that based on the current outputs and input selects a proper sub-model that is valid in that special operating situation. *Regression tree* models may be a viable solution method [de Veaux *et al.*, 1993]. Further incorporation of process knowledge in the modelling may lead to the inclusion of nonlinear rate expressions that can account for the nonlinearities observed.

The LTI batch models introduced in this chapter have large memory requirements and are computationally demanding because the entire batch must be modelled in one model. In most chemical plants and especially the biotechnological systems dealt with in this chapter the sample time allows for plenty of time for the calculations between samples for state estimation, monitoring and control. For other types of real-time applications, e.g. eletronical or mechanical systens, these LTI batch models may be too demanding for the process control system and a more suitable realisation of the model has to be used [Dewilde and van der Veen, 1998].

Differences in sample time for the different variables in the MIMO stacked state space model can easily be handled by adjusting the size of the matrices accordingly as long as the variables are sampled at the same time relative to the batch start. Data sampled at irregular points in time and missing data are common in industrial process operation and such events must be handled if data are to be provided for the model estimation by this equipment. This problem may be more rare in small scale pilot plant equipment. Methods for handling missing data and outliers must be tailored to the stacked state space models taking into account the special model structure and available process knowledge. The handling of such events may be facilitated using state estimators [Soroush, 1998].

5.9 SISO Example

The SISO example introduced in section 5.4.1 will here be described in greater detail.

The model equation for this example is chosen to be simple, yet nonlinear. It is *not* a model of a physical system. The model is described by the ordinary differential equation

$$\frac{dx}{dt} = -x^2 + u, \tag{5.48}$$

where x is the measured state and u is the input to the system. t denotes the time. All units are arbitrary.

The model will be simulated for 10 time units with a sample time of 0.25 time units leading to 41 samples obtained from each batch. For the SISO examples 25 batches will be simulated in order to avoid unwanted effects caused by not having a sufficient amount of data. Initial values of the state and input signal values are presented in table 5.1 and illustrated in figures 5.31 and 5.32.

Various models are estimated and they are validated against a (simulated) validation experiment that is obtained using a special input trajectory different from the trajectories used for estimation of the model. The validation input trajectory is shown in figure 5.33.

5.10 MIMO Example

For a fermentation process we have a simple model that can be used for simulation of the system in fed-batch operation. The model contains only three states: substrate concentration S [g/l], biomass concentration X [g/l] and volume of the fermentor V [l]

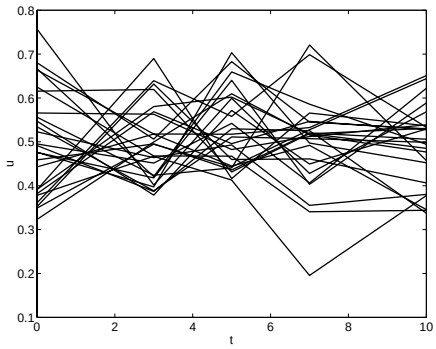


Figure 5.31. Input trajectories $u(t)$ for 25 batches for the SISO simulation example. The values are in table 5.1.

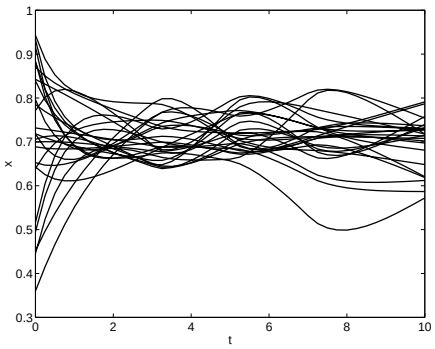


Figure 5.32. Output trajectories $x(t)$ for 25 batches for the SISO simulation example.

Batch	x_0	$u(t=0)$	$u(t=3)$	$u(t=5)$	$u(t=7)$	$u(t=10)$
1	0.79	0.55	0.38	0.57	0.45	0.53
2	0.45	0.39	0.69	0.42	0.57	0.54
3	0.50	0.62	0.47	0.47	0.34	0.34
4	0.45	0.56	0.42	0.44	0.72	0.46
5	0.69	0.46	0.50	0.44	0.52	0.51
6	0.72	0.48	0.42	0.64	0.40	0.59
7	0.88	0.48	0.40	0.70	0.50	0.45
8	0.84	0.67	0.42	0.66	0.59	0.49
9	0.80	0.38	0.57	0.50	0.55	0.53
10	0.52	0.68	0.51	0.68	0.55	0.53
11	0.71	0.53	0.39	0.51	0.51	0.50
12	0.88	0.38	0.47	0.41	0.20	0.38
13	0.84	0.49	0.39	0.53	0.53	0.34
14	0.64	0.32	0.50	0.61	0.52	0.48
15	0.64	0.57	0.56	0.48	0.51	0.54
16	0.65	0.39	0.58	0.60	0.52	0.53
17	0.94	0.62	0.62	0.43	0.52	0.49
18	0.73	0.52	0.46	0.49	0.36	0.38
19	0.92	0.35	0.49	0.44	0.53	0.64
20	0.36	0.36	0.64	0.56	0.70	0.53
21	0.93	0.44	0.52	0.44	0.49	0.35
22	0.70	0.50	0.45	0.54	0.41	0.62
23	0.72	0.35	0.63	0.46	0.46	0.41
24	0.87	0.66	0.52	0.52	0.53	0.65
25	0.77	0.76	0.40	0.60	0.43	0.57

Table 5.1. Values for x_0 and $u(t)$ for the SISO batch example. The simulated values are shown in figures 5.31 and 5.32.

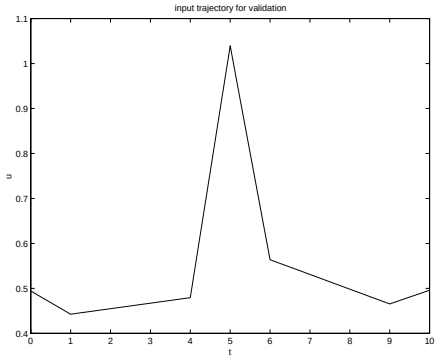


Figure 5.33. Input trajectory for the validation of the SISO model.

$$\dot{S} = -\frac{\mu}{0.5}X + (S_f - S)\frac{F}{V}$$
$$\dot{X} = \mu X - X\frac{F}{V}$$
$$\dot{V} = F,$$

where the growth rate is given by $\mu = \frac{\mu_{max} S}{K+S+0.5S^2}$. The growth rate depends heavily on S. The maximum growth rate is obtained for $S = 0.245$ g/l. For substrate concentrations larger than this value the growth rate slowly decreases. This is illustrated in figure 5.34.

This model is simulated for 4 hours. The nominal initial conditions x_0 are given in table 5.2, but when creating the data sets the initial values will be varied using the specified values as mean values. The input data can be seen in figure 5.35 and the resulting output data can be seen in figure 5.36. The sample time is 0.1 hours leading to 41 samples for the entire batch.

Only the first few model examples of the MIMO stacked state space model are given here. The stacked state space model is given by

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}. \tag{5.49}$$

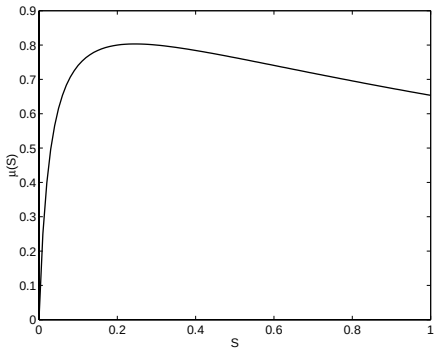


Figure 5.34. Growth rate μ as a function of the substrate concentration S.

Constant	Value
μ_{max}	1
K	0.03
S_f	10 g/l
S_0	0.2 g/l
X_0	1 g/l
V_0	1 g/l

Table 5.2. Initial values and constant parameter values of the model.

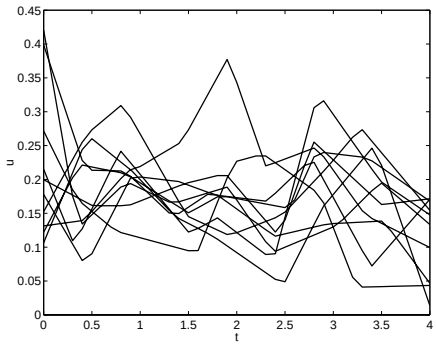


Figure 5.35. Input trajectories used for estimation of the MIMO stacked state space model.

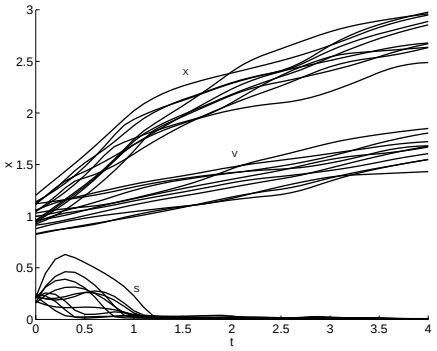


Figure 5.36. Output trajectories used for estimation of the MIMO stacked state space model.

This model type was introduced in section 5.3. Full order (non-causal) models may be estimated leading to dense **A** and **B** matrices. From the result on the SISO system it has been shown that a low order causal model is to be preferred as it has better prediction power. The problem is to find a suitable order of the model.

The **A** and **B** matrices are illustrated for a model with the order $n_a = 15$ and $n_b = 4$ in figures 5.37 and 5.38. This model order can be decreased leading to a more parsimonious model that even have better prediction capabilities for the validation example. A new model is developed with $n_a = 3$ and $n_b = 2$. A contour plot of the model matrices is given in figures 5.39 and 5.40. When the model order is decreased a better illustration of the content of the model matrices can be provided by making a plot of the diagonals of the matrices only. Such diagonal plots for the model matrices in figures 5.39 and 5.40 are shown in figures 5.41 and 5.42 where individual parameters can be seen.

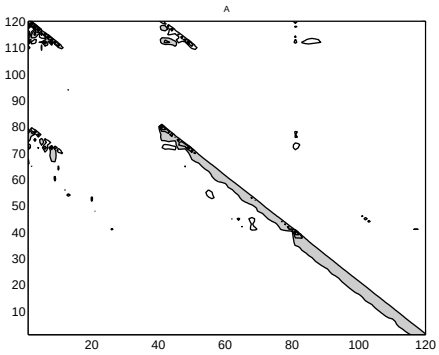


Figure 5.37. **A** matrix for a model with the following model orders: $n_a = 15$ and $n_b = 4$. This contour plot only shows the largest values of the **A** matrix.

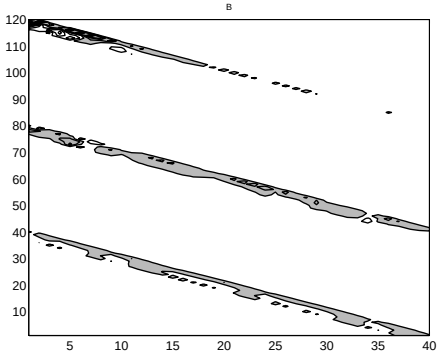


Figure 5.38. **B** matrix for a model with the following model orders: $n_a = 15$ and $n_b = 4$. This contour plot only shows the largest values of the **B** matrix.

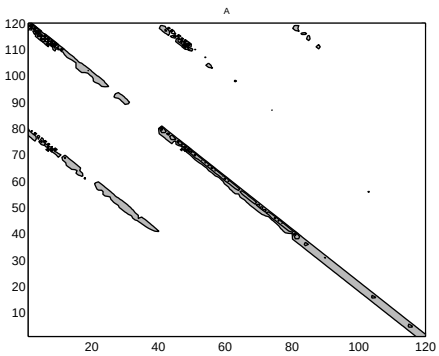


Figure 5.39. **A** matrix for a model with the following model orders: $n_a = 3$ and $n_b = 3$. This contour plot only shows the largest values of the **A** matrix.

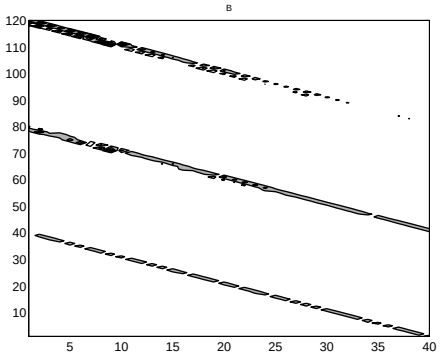


Figure 5.40. **B** matrix for a model with the following model orders: $n_a = 3$ and $n_b = 2$. This contour plot only shows the largest values of the **B** matrix.

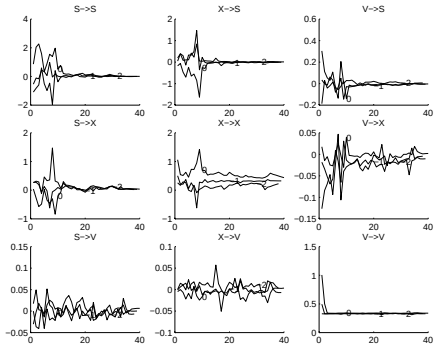


Figure 5.41. A matrix for a model with the following model orders: $n_a = 3$ and $n_b = 3$. This plots only shows the diagonals and subdiagonals of the matrix shown in figure 5.39.

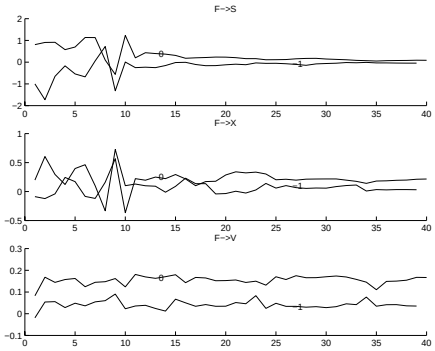


Figure 5.42. B matrix for a model with the following model orders: $n_a = 3$ and $n_b = 3$. This plots only shows the diagonals and subdiagonals of the matrix shown in figure 5.40.

List of Symbols

Letters

Symbol	Description
I	Number of batches in a data set
N	Number of samples from a batch
T	Final time of the batch
T_s	Sample time
a_i	Coefficient on output values for an ARX model
b_i	Coefficient on input value for an ARX model
c_f	Control signal smoothing parameter
$f(x, u)$	Function
$h(x)$	Measurement function
n_a	Order of ARX model. Number of past outputs to include in the model
n_b	Order of ARX model. Number of past inputs to include in the model
n_s	Number of output measurement
n_u	Number of inputs
t	time [h]
u	Input
$u(t)$	Input measurement in deviation variables
x	State or output measurement
x_0	Initial state or output measurement
$\bar{y}$	Average of $\mathbf{y}$
$\mathcal{Y}(t)$	Output measurement
$y(t)$	Output measurement in deviation variables

Matrices

Symbol	Description
$\mathbf{I}$	Identity matrix of suitable dimension
$\mathbf{A}$	Coefficient matrix on output measurements ($n_x n \times n_x n$)
$\mathbf{B}$	Coefficient matrix on input measurements ($n_s n \times n_u n$)
$\mathbf{F}$	Sparse coefficient matrix for parameter estimation. See equation (5.18)
$\mathbf{L}_i$	Discrete approximation to a derivative operator of ith order
$\mathbf{Q}$	Weighting matrix for the optimisation problem
$\mathbf{U}$	Input (actuator) values ($n_u n \times I$)
$\mathbf{X}$	Output measurements ($n_s n \times I$)
$\mathcal{U}(t)$	Input value (n_u)
$\mathbf{u}_k$	Input vector (n)
$\mathbf{u}(t)$	Input vector (n_u)

Symbol	Description
$\mathbf{x}$	Output vector
$\mathbf{x}_k$	Output vector (n)
$\mathbf{x}(t)$	State vector or output measurement (n_s)
$\mathcal{X}(t)$	State value (n_s)
$\dot{\mathcal{X}}(t)$	Derivative of $\mathcal{X}(t)$ (n_s)

Greek Letters

Symbol	Description
α	Limit in equality
δ	Limit in equality
λ	Regularisation parameter
μ	Growth rate $[g/h]$
μ_{max}	Maximum growth rate, 1.0 g/h
$\boldsymbol{\theta}$	Parameter vector

Symbols

Symbol	Description
$\|\mathbf{A}\|$	The 2-norm of $\mathbf{A}$
$\mathbf{A}^{\#}$	Pseudo inverse of $\mathbf{A}$
∂	Partial derivative

6

Dynamic Batch Process Monitoring using Local Linear Models

This paper bridges process chemometrics and dynamic modelling of batch processes. A stacked state space model has previously been developed. It is a local linear state space model developed especially for batch processes. It is shown that correlation models used for batch process monitoring is a superset of stacked state space models where the latter has been restricted with an explicit formulation of a dynamic, causal model structure. Stacked state space models can be used for monitoring just like the correlation models with the addition that the task can be subdivided into monitoring the prediction capability of the dynamic model and allows monitoring of the inputs and outputs separately.

6.1 Introduction

This chapter shows that process chemometrics models used for fault diagnosis of batch processes can be related to dynamic time series models used to model batch processes. The chapter highlights the similarities and differences between the two modelling types. The largest difference is that process chemometric models such as PCA and PLS are noncausal unstructured models while time series models are causel type models.

Process chemometric methods described in [MacGregor and Nomikos, 1992; Nomikos and MacGregor, 1995; Martin *et al.*, 1996; Gregersen and Jørgensen, 1999] are powerful tools for analysing process data and for constructing models that can be used for fault diagnosis.

Process chemometrics have been applied to many different processes with good result [Martin *et al.*, 2002; Kourti, 2002; Undey and Cinar, 2002]. In the literature the methods are often applied to polymerisation reactors or biochemcal processes, but the methods are quite general and may be applied to any batch process in order to verify that the process is running according to the

prescribed recipe.

Modelling using process chemometrics is often termed *data driven modelling* since these methods are often applied without too much influence by prior knowledge, but rather with the intent that the data should be used to describe the system behaviour. For fault diagnosis a data driven approach may be desired since a knowledge based approach can be very time consuming because one has to model the system at hand not only in its normal condition, but also include faulty modes in the model thus possibly leading to a very large and complex model.

Fault diagnosis systems may include data driven methods as well as knowledge based methods [Chiang *et al.*, 2001]. Knowledge based method have traditionally been used in critical systems such as airplanes, but even for such systems data based methods are used when necessary [Huo *et al.*, 2001].

Process chemometric models used for fault diagnosis are rarely used for prediction purposes and may therefore be based on process data obtained from normal operation.

An extension of conventional ARX models to include dynamic models for batch processes is presented in [Gregersen and Jørgensen, 2002]. This model type can be used for simulation of the process in order to perform e.g. what-if analyses, control design and as it will be shown in this paper, for fault diagnosis. Identification of these dynamic models have significant requirements on the sampling interval for the measurements depending on the process dynamics. Aliasing must be avoided by filtering high frequency content of the signals before sampling. The most challenging part of dynamic modelling is that the system must be persistently excited for the model to be identifiable. This requires special experiments that may be costly to perform since they may severely affect the process outcome when state trajectories are changed.

This paper will show that dynamic models can be used for batch monitoring and that the developed dynamic models have some similarities with the static models conventionally used for process chemometrics. It is important the right type of model is selected depending on the task at hand. If the problem is mainly related to process montoring then it will be a smaller challenge to retrieve process data from a normally opetating process and estimate a traditional chemometric model than to obtain highly dynamic data from several experiments in to indentify the more complex dynamic batch models.

Process chemometrics modelling is presented in section 6.2 with a motivation for using dynamic models for process chemometric applications. A short introduction to ARX modelling is given in section 6.3 and the necessary extension to batch processes is given in section 6.4. Here also a simple estimation algorithm for the model parameters is presented. A short presentation of the equations used to perform faults diagnosis using PCA is given in section 6.5. A comparison and integration of process chemometrics and process dynamics is performed in section 6.5.1 and a proof of concept example is given in section 6.6. The discussion and conclusions are given in section 6.7.

6.2 Chemometrics and process control

For practical model development one often has the option of choosing *either* a data based approach *or* a knowledge based approach. The combination of these two methods is of course much more powerful than considering the two approaches separately. The difficulty lies in how to combine knowledge and data *systematically* such that the available information is utilised to its fullest. By incorporating process dynamics knowledge into the data driven modelling one can obtain a model structure and model parameters that are more transparent and thereby closer to the behaviour of the real system. Hence, a model may be obtained that offers insight into the operation and control of the plant which is often desired when modelling [Rusell *et al.*, 2000].

When combining knowledge and data into the same framework there lies a danger of misrepresentation. It may be that either the process knowledge does not reflect the real (and possible complex) system at all times. This may especially be the case under faulty conditions. It may also be the case that data used for modelling is misrepresentative of the real plant behaviour. E.g. due to sensor bias or noise and the dynamics of the sensors.

This paper combines knowledge from two different areas. Process chemometric and process systems engineering. Process chemometrics is dealt with in a large number of papers e.g. [Nomikos and MacGregor, 1995; MacGregor, 1997; Kourti, 2002; Martin *et al.*, 1996, 2002; Wise and Gallagher, 1996].

The chemometric methods for process analysis and fault diagnosis dealt with in [Nomikos and MacGregor, 1995; Martin *et al.*, 1996] have radically different origins than the methods for linear dynamic time series modelling of batch processes presented in [Gregersen and Jørgensen, 2002]. Chemometric methods result from explorative data analysis whereas dynamic models have their traditional origin from process control and systems analysis.

The relationship between state space models for continuous systems and chemometric tools was investigated in Wise [1991]. PLS, CVA (canonical variate analysis) and subspace methods have been used as numerical tools for solving the parameter estimation problem when developing ARX and state space models for stationary dynamic systems [Simoglou *et al.*, 2002; Juricek *et al.*, 1999; Schaper *et al.*, 1990].

An introduction to state space models can be found in [Rugh, 1996] that emphasises on classic control theory, stability analysis etc. A review of continuous-time identification is given in [Unbehauen and Rao, 1998]. Ljung [1987] presents the classical introduction to discrete linear time-invariant and time-varying input/output modelling.

Modelling of batch processes using input/output models is not covered in the literature in great detail. However, [Russell *et al.*, 1998] discusses the need for improved models for improved batch process monitoring and operation. The book [Dewilde and van der Veen, 1998] discusses I/O modelling of time-varying systems in great detail; however not for batch processes, but for systems that run indefinitely. This book is mostly devoted to stability analysis and com-

putational efficiency of the developed algorithms and is therefore not directly applicable to batch processes. The use of subspace identification methods on time-varying systems is covered in [Verhaegen and Yu, 1995; Liu, 1997], but is limited to stationary systems. The use of PLS and CVA (canonical variate analysis) for modelling continuous process dynamics is treated in [Simoglou *et al.*, 2002; Negiz and Çinar, 1998; Çinar and Undey, 1999].

Reviews of control of batch processes in general is given in [Berber, 1996]. The example in this paper is for a simulated fermentor. The control of fermentors is reviewed in [Rani and Rao, 1999; Shimizu, 1993; Jørgensen and Jensen, 1989].

In the following an introduction to linear time series models is given. This model type is then extended to cover time-vaying system with some adaptations to cover batch processes.

6.3 ARX for continuous processes

An ARX model for a continuous system can be written

$$\begin{aligned} y(k) = a_1 y(k-1) + a_2 y(k-2) + \cdots + a_{n_a} y(k-n_a) \\ + b_1 u(k-1) + b_2 u(k-2) + \cdots + b_{n_b} u(k-n_b), \end{aligned} \quad (6.1)$$

where $y(k)$ and $u(k)$ are the outputs and inputs at time k. The model parameters a_k and b_k are constant for time-invariant systems. A shorter version of this model is

$$y(k) = \Theta^\top \phi(k), \quad (6.2)$$

where Θ contains the parameters that must be identified from data and $\phi(k)$ contains previous outputs and inputs

$$\phi(k) = \big[y(k-1)\ y(k-2)\ \ldots\ y(k-n_a)\ u(k-1)\ u(k-2)\ \ldots\ u(k-n_b) \big]. \quad (6.3)$$

Using a least squares formulation to estimate the parameters we obtain

$$\Theta = (\phi_k^\top \phi_k)^{-1} \phi_k \mathbf{y}_k. \quad (6.4)$$

Since the matrix $(\phi_k^\top \phi_k)$ contains variables that are correlated and the time shifting of the variables introduces additional correlation between elements in the matrix ϕ_k the matrix inversion in equation (6.4) can be difficult to perform due to a high condition number. Hence, suitable numerical methods for solving the system of equations must be chosen [Golub and van Loan, 1991].

When the parameters in this model type are identified using PCA it is called Dynamic PCA [Wise, 1991; Simoglou *et al.*, 2002; Juricek *et al.*, 1999; Luo *et al.*, 1999]. When PLS is used it is called Dynamic PLS [Simoglou *et al.*, 1999; Lakshminarayanan *et al.*, 1997]. However, the model type remains an ARX model independently of the numerical method used to solve the parameter estimation problem.

6.4 Stacked state space models.

For batch processes conventional ARX models are not suitable due to the nonlinear, nonstationary behaviour of the batch process.

The goal of this section is to describe a method for modelling batch processes using a linear time invariant model (LTI models). This can be seen as very challenging since batch processes conventionally are modelled as nonlinear ordinary differential equations (ODE), which lead to time-varying models when linearised. The challenge is handled by creating local linear models of the system and stack them on top of each other such that an individual model exists for each sample instant. By stacking the model into a single structure a linear state space model is finally obtained that is called a stacked state space model, which is short for finite-horizon, time-shifted, stacked, linear state space models. The steps to create the model is given in the next section and a more detailed description of this methodology may be found in [Gregersen and Jørgensen, 2002].

A LTI-model of a batch process can be built if it is possible to linearise the model around a normal operating trajectory. This is often the case for industrial batch processes. If the variations of the process around the trajectory are not too large a linear model will give satisfactory precision. In order to make LTI models for batch processes a input/output formulation is developed. This paper will only describe the case where there is a single input and a single output (SISO). The multi-input/multi-output case is presented in [Gregersen and Jørgensen, 2002]. Based on (6.1) a time-varying ARX model can be formed

$$\begin{aligned} y(k) = a_1(k)y(k-1) + a_2(k)y(k-2) + \cdots + a_{n_a}(k)y(k-n_a) \\ + b_1(k)u(k-1) + b_2(k)u(k-2) + \cdots + b_{n_b}(k)u(k-n_b), \end{aligned} \quad (6.5)$$

where the a and b parameters are time-varying to account for the change of dynamics as the batch progresses. This is illustrated in figure 6.1 where is can be seen that the batch process can be modelled by changing the model at every sampling instant. In order to obtain a combined model the inputs and outputs can be stacked in order to obtains a stacked model. This will be shown in the following.

A new output vector $\mathbf{x}$ is defined. This vector contains all measurements from the entire batch. This assumes equal duration of the batches which is also a common goal in industrial batch processing where it is often attempted to follow a fixed recipe. Thus the new output vector is:

$$\mathbf{x} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad (6.6)$$

where y_i is $y(t = T_s i) = y(k = i)$. The sample time T_s is assumed constant.

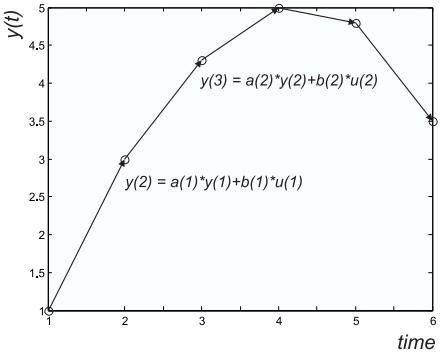


Figure 6.1. The stacked state space model can be viewed as a time varying ARX model where each row in the model matrices correspond to certain time instant. The challenge is to find a way to estimate all the parameters in the model simultaneously.

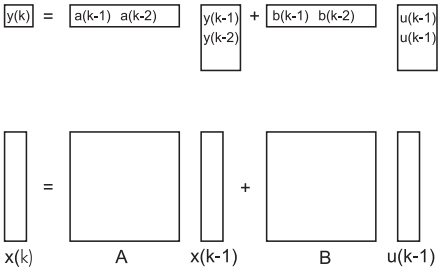


Figure 6.2. Graphical representation on the model matrices for the stacked state space model. The illustration on the top shows the structure of a single ARX model. These models are stacked on top of each other to obtain the stacked state space model shown in the lower part of the figure. Using this model structure a model of the entire batch can be estimated in a single procedure.

Past outputs as well as inputs are used to estimate new outputs:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}, \tag{6.7}$$

where $\mathbf{A}$ and $\mathbf{B}$ are the model matrices that define the dynamics of the system and the data vectors are defined by $\mathbf{x}_k = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^\top$, $\mathbf{x}_{k-1} = \begin{bmatrix} y_0 & y_1 & \cdots & y_{n-1} \end{bmatrix}^\top$ and $\mathbf{u}_{k-1} = \begin{bmatrix} u_0 & u_1 & \cdots & u_{n-1} \end{bmatrix}^\top$.

6.4.1 Estimation

It is assumed that data are collected from several batch runs using different input trajectories during each of the batches. The output values are collected in the matrices $\mathbf{X}$ $(n+1 \times I)$ and $\mathbf{U}$ $(n+1 \times I)$, where I is the number of batch runs. Column i of the matrices contains data from the ith batch. The number of batch runs will usually be (much) less than the number of sampled data points $(n+1)$. For modelling purposes submatrices of $\mathbf{X}$ and $\mathbf{U}$ are defined that contain the vectors $\mathbf{x}_k$ and $\mathbf{u}_{k-1}$ from different batches, respectively.

For finding a solution to the model $\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{B}\mathbf{U}_{k-1}$ we rewrite the equations into

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{k-1} \\ \mathbf{U}_{k-1} \end{bmatrix}, \tag{6.8}$$

The solution to this problem can be stated as a least squares problem

$$\min_{\begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}} \left\| \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix} \begin{bmatrix}\mathbf{X}_{k-1} \\ \mathbf{U}_{k-1}\end{bmatrix} - \mathbf{X}_k \right\|. \tag{6.9}$$

A solution to this problem can be found using the pseudo inverse

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix} = \mathbf{X}_k \begin{bmatrix}\mathbf{X}_{k-1} \\ \mathbf{U}_{k-1}\end{bmatrix}^{\#} \tag{6.10}$$

6.4.2 Causality Constraints

By imposing structure on the matrices based on causality and other process knowledge the number of elements to be estimated in the model matrices can be significantly reduced from the general case presented in the previous section.

For state space models of the type $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}$ some constraints can be imposed if a causal model is desired. The developed models so far all have elements above and below the diagonal in the $\mathbf{A}$ and $\mathbf{B}$ matrices. In order to ensure a causal model only non-zero elements can be permitted on and below the diagonal in both model matrices. Based on physical knowledge the absolute value of the entries in the matrices should decrease as they get farther away from the diagonal as this indicates interaction between measurements that are far away in time.

To obtain a causal model equation (6.7) is rewritten with explicit elements where the new causal structure of the model is introduced

$$\begin{bmatrix}x_1 \\ x_2 \\ x_3\end{bmatrix} = \begin{bmatrix}a_{11} & \dots & \mathbf{0} \\ a_{21} & a_{22} & \vdots \\ a_{31} & a_{32} & a_{33}\end{bmatrix} \begin{bmatrix}x_0 \\ x_1 \\ x_2\end{bmatrix} + \begin{bmatrix}b_{11} & \dots & \mathbf{0} \\ b_{21} & b_{22} & \vdots \\ b_{31} & b_{32} & b_{33}\end{bmatrix} \begin{bmatrix}u_0 \\ u_1 \\ u_2\end{bmatrix}. \tag{6.11}$$

The challenge is to estimate the non-zero elements in the matrices $\mathbf{A}$ and $\mathbf{B}$ while maintaining the value zero where intended due to causality. The model (6.11) can be rewritten into

$$x_1 = a_{11}x_0 + b_{11}u_0 \tag{6.12}$$
$$x_2 = a_{21}x_0 + a_{22}x_1 + b_{21}u_0 + b_{22}u_1 \tag{6.13}$$
$$x_3 = a_{31}x_0 + a_{32}x_1 + a_{33}x_2 + b_{31}u_0 + b_{32}u_1 + b_{33}u_2, \tag{6.14}$$

where it is immediately clear that outputs are only affected by past outputs and inputs. These equations result in the following equation system where the

parameters to be estimated are extracted as a vector

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_0 & & & & & & u_0 & & & & & \\ & x_1 & & x_0 & & & & u_1 & & u_0 & & \\ & & x_2 & & x_1 & x_0 & & & u_2 & & u_1 & u_0 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{22} \\ a_{33} \\ a_{21} \\ a_{32} \\ a_{33} \\ b_{11} \\ b_{22} \\ b_{33} \\ b_{21} \\ b_{32} \\ b_{33} \end{bmatrix} \tag{6.15}$$

$$\mathbf{x}_3 = \mathbf{F}\boldsymbol{\theta}, \tag{6.16}$$

where $\mathbf{F}$ contains the measured input and output data. Note that $\mathbf{F}$ in general will be sparse and a suitable sparse solver must be chosen to obtain the parameters in an efficient manner.

6.4.3 Model Order

For simple ARX models the notion of model order describes the number of past sampled values that should be included in the model. The model order is described by the values n_a and n_b introduced in equation (6.1). The optimal model order is determined by the sample time, the dynamics of the system, the frequency spectrum of the input signal etc. In reality the system order will also be affected by the signal to noise ratio of the data and the number of samples and batches available. In practice n_a and n_b can be time-varying. However here they are assumed constant throughout the batch.

For the stacked state space models the order of the ARX model can be transformed into having a model where only the diagonal and a few sub-diagonals are non-zero. The number of non-zero diagonals and sub-diagonals is the order of the model.

6.5 Process Chemometrics for Fault Diagnosis

The use of process chemometric methods for fault diagnosis dates back to [Kresta *et al.*, 1991]. This article has been followed up by [MacGregor and Nomikos, 1992; Nomikos and MacGregor, 1995] and many others. Contributions are also given in [Martin *et al.*, 1996; Martin and Morris, 1996; Albert *et al.*, 1997]. Recent reviews are given in [Wise and Gallagher, 1996; Martin *et al.*, 2002; Kourti, 2002; Çinar and Undey, 1999; Undey and Cinar, 2002].

Recent examples of applications of process chemometrics for fault diagnosis can be found in [Lennox *et al.*, 2000; Tates *et al.*, 1999; Neogi and Schlags, 1997; Gregersen and Jørgensen, 1999].

All of the articles mentioned above use the same data structures and essentially the same numerical methods for performing the parameter estimation in the models. The methodology based on PCA will be presented in the following.

Batch process data is often stored in a three-way matrix as shown in figure 6.3. In order to submit the data to process chemomcatrical methods such as PCA or PLS the matrix are unfolded to obtain 2D matrices.

6.5.1 Process Chemometrics Methods Combined with Stacked State Space Models.

One may observe that the way data matrices are handled when doing PCA and PLS multiway modelling is very similar to the handling of the matrices when performing dynamic time series modelling. Both model types are examples of unfolding the underlying three-way matrix into the more comfortably handled two-way matrix as illistrated in figure 6.3. The two areas *process chemometrics* and *dynamic modelling* have different background and immediate purpose and therefore provide different obvious ways to unfold the data resulting in apparently different models of the system.

For the fault diagnosis application there is a significant emphasis on the time

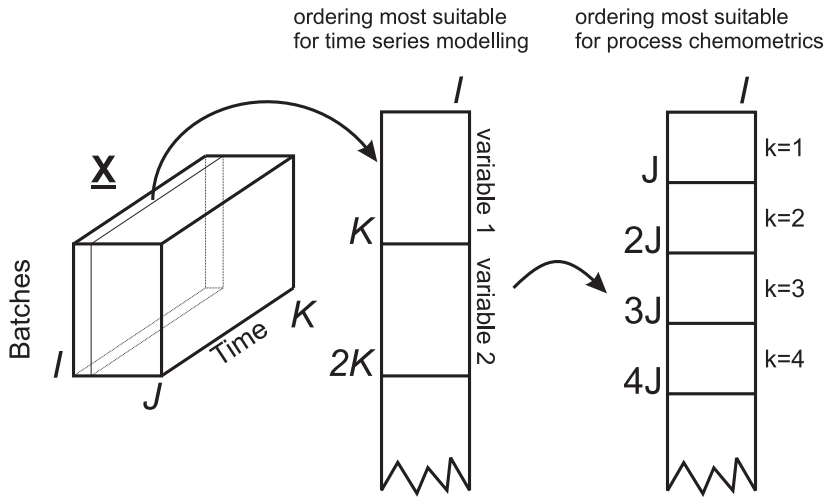


Figure 6.3. Visualisation of the unfolding of a three-way matrix used to store batch process data. Reordering scheme during unfolding of the three-way data matrix. Depending on the application of the data, different ways of unfolding may be obvious. I denotes the number of batches, J denotes the variables and K denotes time.

aspect. By grouping data such that data that are measured at the same sample time instant matrix handling is facilitated when the batch evolves and more data are processed as illustrated on the right hand side of figure 6.3. Only once a fault has been detected does it become important to investigate individual variables and their relationships. For the dynamic modelling application the emphasis is on the evolution of the variables. For times series models the plots of parameters and the specification of regularisation parameters and structure become much easier to read when the grouping of data is by variable, so all measurements of the same variable (at different points in time) are grouped together. It is important to note that the different ways of unfolding are simply ways to display data and how index the elements. The data considered is the same and the choice of unfolding is simply a matter of convenience.

Figure 6.3 illustrates the different orderings of the unfolded matrices used for fault diagnosis and LTI modelling. It is directly seen that the data matrices are the same except for a reordering of rows. For a PCA this lead to a reordering of the rows of the scores, but loadings remain unchanged from the change of row order. The two models are in a one to one relationship and can be converted to each other simply by observing the appropriate row order.

The unfolded data matrix used for MPCA modelling in this section will be be termed $\mathbf{D}$. This matrix contains the same data as the data matrix used for the stacked state space modelling. The reordering of rows will be left silently out of the presentation. It must also be noted that for the time series model the variables must be divided into inputs and outputs. This is not necessary for the MPCA models.
Using the SVD the data matrix is decomposed into

$$\mathbf{D} = \begin{bmatrix} \mathbf{X}_{k-1} \\ \mathbf{U}_{k-1} \end{bmatrix}^\top = \mathbf{U}\mathbf{M}\mathbf{V}^\top = \mathbf{T}\mathbf{P}^\top. \tag{6.17}$$

This is the same as performing a PCA on $\mathbf{D}$ where the scores are $\mathbf{T} = \mathbf{U}\mathbf{M}$ and the loadings are $\mathbf{P} = \mathbf{V}$ [Jackson, 1991]. See appendix A.4 for a description of the SVD.

Recall equation 6.8, which describes the linear *noncausal* dynamic behaviour:

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{X}_{k-1} \\ \mathbf{U}_{k-1} \end{bmatrix},$$

where the last matrix is equal to the matrix $\mathbf{D}^\top$. It is then seen that

$$\mathbf{X}_k^\top = \begin{bmatrix} \mathbf{X}_{k-1} \\ \mathbf{U}_{k-1} \end{bmatrix}^\top \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}^\top \tag{6.18}$$

$$= \mathbf{U}\mathbf{M}\mathbf{V}^\top \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}^\top. \tag{6.19}$$

Using this last equation we now have the possibility to define a new set of scores and loadings for the *output* matrix $\mathbf{X}_k^\top$. The new loadings become

$$\tilde{\mathbf{P}} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix} \mathbf{V}. \tag{6.20}$$

A new set of scores for the outputs can be calculated using these loadings

$$\tilde{\mathbf{T}} = \mathbf{X}_{k-1}\tilde{\mathbf{P}}. \tag{6.21}$$

It is seen that using the model matrices $\mathbf{A}$ and $\mathbf{B}$ as filtering matrices on the original formulation of the MPCA (and MPLS) problem a new model structure emerges where loadings and scores are based not only on the projection given by the PCA analysis, but also on the identified dynamic time series model.

6.5.2 Monitoring

The separation of variables into inputs and outputs and the development of stacked state space model allow us to develop the following new strategy for monitoring of the plant. A four step fault diagnosis methodology for batch processes where the measurements are separable into inputs and outputs is proposed

1. Diagnosis of the input variables. All plants have bounds on actuators. Safety bounds may be imposed by the design of the process. Furthermore when several actuators are available on a plant a MSPM scheme may be set up for the actuators alone.

2. Diagnosis of the dynamic model validity based on the prediction error of the model [Russell *et al.*, 1998; Chiang *et al.*, 2001]. Using standard SPM techniques such as CUSUM or EMWA the model/system mismatch can be monitored. The residuals will be non-zero due to faults, disturbances, noise and/or modelling errors. Also for this case multivariate monitoring may be applied.

3. If the dynamic model *can* be applied monitoring of the output variables using the dynamic batch model loadings (6.20) and scores (6.21) can be used.

4. If the dynamic model *can not* be applied due to system/model mismatch or process faults a fall back strategy using the fault diagnosis model in [Gregersen and Jørgensen, 1999].

Dividing the monitoring into 4 tasks makes it easier to diagnose in the case of a fault. Firstly, the data is separated into inputs and outputs for two reasons. Simple bounds are easy to set up for inputs and is always implemented on industrial process plants. Secondly, the distinction into inputs and outputs makes it easier to isolate the cause of faults since the model is causal. Monitoring the prediction errors is a way to detect if the dynamics of the system has changed during the batch. E.g. changes in reaction rates or heat transfer rates can be detected. As long as the dynamic model is valid this method is not sensitive to changes in the mean of the process trajectory. Fed-batch processes are inherently integrating processes and disturbances may therefore persist throughout the batch.

The last two steps are performed depending on the validity on the dynamic model. If the dynamic model is valid this model will give the best predictions of the batch behaviour. Since the inputs at this time already have been subject to the monitoring it is only necessary to investigate the outputs.

In the event of detection of a fault using the dynamic input/output scheme we may have to use a conventional static batch monitoring model as a fall-back strategy. If this is ever necessary a fault has at this point been identified using the dynamic model, but in cases where faults are severe the dynamic model may not necessarily be useful to isolate the fault.

6.6 Example

For a fermentation process a simple nonlinear model can be used for simulation of the system in fed-batch operation. The model contains only three states: substrate concentration S [g/l], biomass concentration X [g/l] and volume of the fermentor V [l]

$$\begin{aligned}\dot{S} &= -\frac{\mu}{0.5}X + (S_f - S)\frac{F}{V}\\ \dot{X} &= \mu X - X\frac{F}{V}\\ \dot{V} &= F,\end{aligned}$$

where the growth rate is given by $\mu = \frac{\mu_{max}S}{K+S+0.5S^2}$. The growth rate depends strongly on S. The maximum growth rate is obtained for $S = 0.245$ g/l. For substrate concentrations larger than this value the growth rate slowly decreases. This is illustrated in figure 6.4.

6.6.1 Model estimation.

The model is simulated for 4 hours. The nominal initial conditions x_0 are given in table 6.1, but when creating the data sets the initial values will be varied using the specified values as mean values. These values are shown in in figure 6.5. The sample time is 0.1 hours leading to 41 samples for the entire batch.

We will not here go into detail about the estimation of the model and selection of the model order since this is described in the paper [Gregersen and Jørgensen, 2002]. For the selected problem the model order for the outputs and the input to be $n_a = 2$ and $n_b = 1$ respectively. When estimating this model it has explicitly been included that there is no direct influence from S and X on V. This leads to a model with parameters shown in figures 6.6 and 6.7. Since the **A** and **B** are sparse only the diagonal and sub-diagonals are plotted. This makes it easy to view all the model coefficients and see how the parameters change with time.

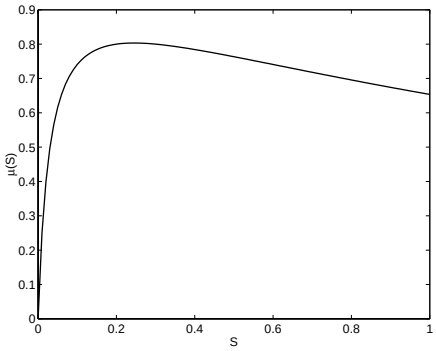


Figure 6.4. Growth rate μ as a function of the substrate concentration S.

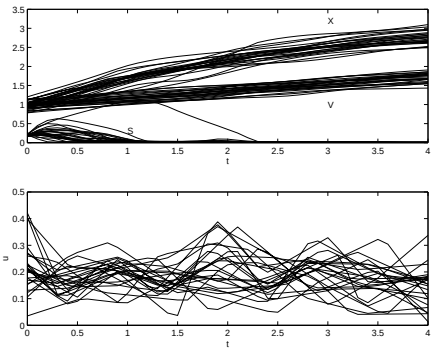


Figure 6.5. Input and output trajectories used for estimation of the MIMO stacked state space model.

Constant	Value
μ_{max}	1
K	0.03
S_f	10 g/l
S_0	0.2 g/l
X_0	1 g/l
V_0	1 g/l

Table 6.1. Initial values and constant parameter values of the model.

Since there are 3 outputs and one input the model matrices are block matrices. The low order of the model results in each block being a diagonal matrix. The $\mathbf{A}$ matrix has furthermore entries in the subdiagonal since $n_a = 2$. The values of the diagonal and subdiagonal are shown in figures 6.6 and 6.7. It is clear that the parameters vary with time.

The model predicts the system outputs reasonably well. The biomass concentration and volume is estimated very well in an open loop simulation whereas the substrate concentration is predicted with less accuracy when the system is simulated with a new substrate feed rate profile shown in figure 6.8. It must be noted that F for this simulation has a peak value that is higher than the values in the data used for model estimation. This peak caused the substrate concentration to be underestimated for the duration of the peak due to model/system mismatch.

6.6.2 Monitoring

A simulation is performed where faults are introduced into the process operation. In the interval from $t = 1$ to $t = 1.5$ an increased amount of substrate

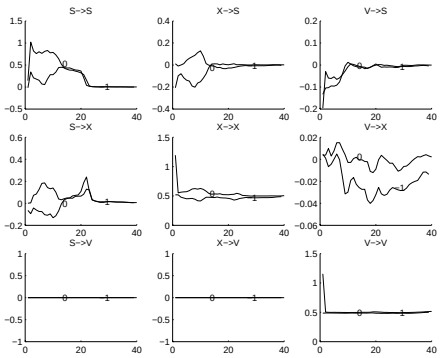


Figure 6.6. Plot of the diagonal and subdiagonal of the **A** matrix for the estimated stacked state space model.

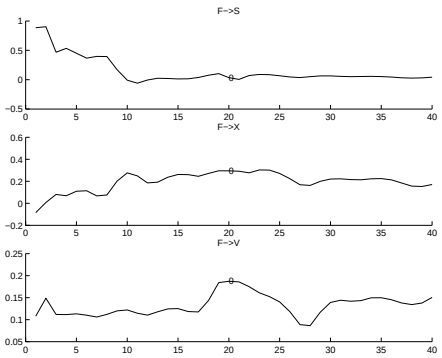


Figure 6.7. Plot of the diagonal of the **B** matrix for the estimated stacked state space model.

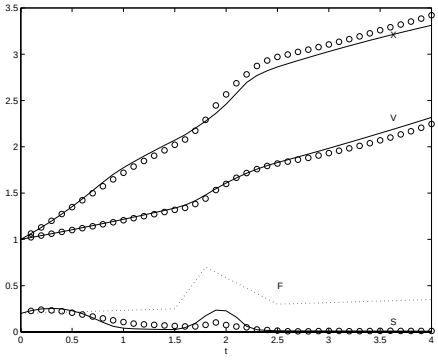


Figure 6.8. Pure simulation for validating the estimated model. Substrate feed rate profile and the corresponding output from the system (-) and model (∘).

is fed to the fermentor. Furthermore a fault is introduced in the dynamics of the system. At $t = 2.5$ the maximum growth rate is reduced from $\mu_{max} = 1$ to $\mu_{max} = 0.8$ simulating a severe inhibition of growth potential. The resulting output are shown in figure 6.9.

Using the CUSUM method based on 1-step prediction errors of the individual outputs the plot shown in figure 6.10 is obtained. The plot shows clearly the events occurring at $t = 1.5$ and $t = 2.5$ as sharp bends of the curves.

6.7 Conclusion

The introduction of a new dynamic model type for fault diagnosis allows the user to incorporate process knowledge and causality constraints for processes where such knowledge is available. The requirements are that the measurements are separable into inputs and outputs and that a suitable dynamic model can be identified. However, this is not always possible for a process running under normal operating conditions and designed experiments may be necessary to obtain sufficient information about the dynamics of the process in order to perturb all inputs sufficiently such that the input/output mapping can be identified.

The stacked state space model is developed by explicitly separating variables into inputs and outputs. By imposing process knowledge such as causality on the model, a model type that is a subset of the conventional process chemometrics models, is obtained. The unstructured process chemometric models may be an advantage for some systems where prior knowledge of the system is not very accurate. When imposing model structure and other assumptions on process modelling it is essential that constraints reflect the actual system and operation. The choice of which model type to select depends on the system and the

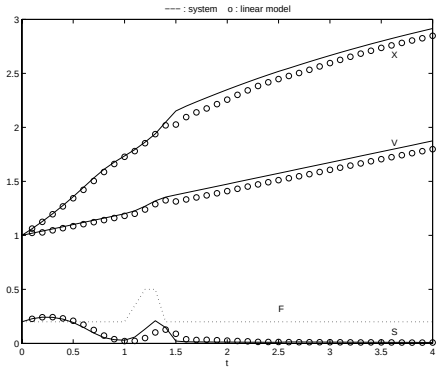


Figure 6.9. Substrate feed rate profile and the corresponding output from the system (-) and model (∘).

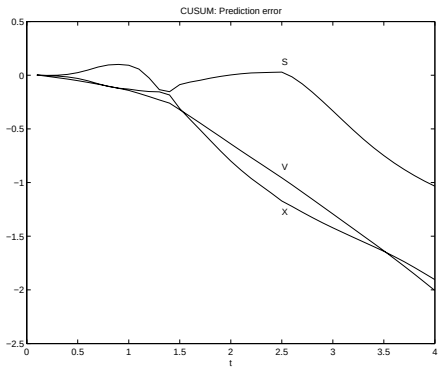


Figure 6.10. CUSUM plot for the 1-step prediction errors.

purpose of the model as well as the available measurements and the operation of the plant. The MPLS and MPCA methods conventionally used for Multivariate Statistical Process Monitoring (MSPM) have a long history of applications where it is shown that these models are highly applicable and useful [Martin *et al.*, 2002; Kourti, 2002; Undey and Cinar, 2002; Gregersen and Jørgensen, 1999]. Adding additional measurements or performning time-consuming experiments may be prohibitively expensive and time consuming and the exitation of the input signals in order to identify the dynamics may change the process behaviour if the process is disturbed too much.

This paper has demonstrated the similarities between the MPCA and MPLS methods conventionally used for MSPM and the stacked state space models used for dynamic I/O-modelling of batch processes. It has been shown how to explicitly introduce a dynamic causal modelling element into the model structure. For the conventional process chemometrics models the information about the evolution in time of the process is omitted and the PCA and PLS algorithms are trusted to account for the dynamics of the system. It has been shown that these models can be compared to *non-causal* black-box models. Dynamic PLS modelling is one way of introducing the dynamics explicitly for stationary processes while stacked state space models is the extension to highly time-varying processes such as batch processes.

The overall succes of the proposed procedure for fault diagnosis using the stacked state space models depends on the prediction capability of the model under normal operating conditions. If the model is poor the prediction error will be large even under normal operating conditions. Hence, it will be difficult to determine a suitable detection level for the faults that will give an accurate detection of actual faults in the system. Further work in this direction is necessary in order to find the most reliable statistics for detection of error conditions and suitable detection levels.

7

A Conceptual Solution to the Generic Fed-batch Control Problem.

Dynamic linear time invariant (LTI) models constitute an important model type applied in process control and optimisation. This linear model type is important because it is straightforward to develop the models (from e.g. data or nonlinear models) and to apply the models for process control and process optimisation. Linear models can be used for local analysis of the underlying plant in many cases even when the process is nonlinear.

Batch processes are often described using nonlinear and/or time-varying models, but it is shown that linear time-invariant models can be constructed either from first principles models or from process data. The model type is called a stacked state space model.

This paper shows how the developed models may be applied to process optimisation and model predictive control.

7.1 Introduction

This paper shows how linear time-invariant models presented in [Gregersen and Jørgensen, 2002] can be used for optimisation and control of batch processes. The contribution of this paper is to show that batch process models that may be identified directly from process data can be introduced in process optimisation and process control schemes. Batch and fed-batch processes require special attention due to their finite duration. Hence, the initial and terminal conditions must be taken care of in the derivation and implementation of the algorithms. An example is given using a simulated fed-batch fermentation.

Modelling of batch processes using input/output models is not covered in the literature in great detail. However, some information can be found in [Russell *et al.*, 1998] that discusses the need for improved models for improved batch process monitoring and operation. The book [Dewilde and van der Veen, 1998] dicusses I/O modelling of time-varying systems in great detail; not for batch

processes, but for systems that are not limited in time. This book is mostly devoted to stability analysis and computational efficiency of the developed algorithms and is therefore not directly applicable to batch processes. The use of subspace identification methods on time-varying systems is covered in [Verhaegen and Yu, 1995; Liu, 1997], but is limited to stationary systems.

Reviews of control of batch processes in general is given in [Berber, 1996]. The example in this paper is for a simulated fermentor. The control of fermentors is reviewed in [Rani and Rao, 1999; Shimizu, 1993].

The models used in this paper can be found in [Gregersen and Jørgensen, 2002]. That paper describes in great detail how models of batch processes may be *identified* from process data. In the present paper just the equations necessary for understanding the optimisation and control aspects are given. In the event that a first principles model of the system is available the models used in this paper may be derived simply by linearisation of the model around the operating trajectory.

The paper starts with a brief definition of model structure and symbols in section 7.2. Operational aspects such as optimisation and optimising control is addressed in section 7.4. Model predictive control (MPC) is treated in section 7.5 and a special proportional controller is also presented in this section. An example is given in section 7.6 using a simulated data from a simple fed-batch fermentation process with three measurements. Discussion and conclusions are presented in section 7.7

7.2 Modelling Batch Processes for Control

7.3 Stacked state space models

For batch processes conventional ARX models are not suitable due to the nonlinear, nonstationary behaviour of the batch process.

The goal of this section is to describe a method for modelling batch processes using a linear time invariant model (LTI models). This can be seen as very challenging since batch processes conventionally are modelled as nonlinear ordinary differential equations (ODE), which lead to time-varying models when linearised. The challenge is handled by creating local linear models of the system and stack them on top of each other such that an individual model exists for each sample instant. By stacking the model into a single structure a linear state-space model is finally obtained. This model type is called a stacked state space model, which is short for finite-horizon, time-shifted, stacked, linear state space models. The steps to create the model is given in the next section and a more detailed description of this methodology may be found in [Gregersen and Jørgensen, 2002].

A LTI-model of a batch process can be built if it is possible to linearise the model around a normal operating trajectory, which is often the case for industrial batch processes. In order to make LTI models for batch processes a

input/output formulation is developed. This paper will only describe the case where there is a single input and a single output (SISO). The multi-input/multi-output case is presented in [Gregersen and Jørgensen, 2002]. A time-varying ARX model is formed

$$y(k) = a_1(k)y(k-1) + a_2(k)y(k-2) + \cdots + a_{n_a}(k)y(k-n_a) + b_1(k)u(k-1) + b_2(k)u(k-2) + \cdots + b_{n_b}(k)u(k-n_b), \quad (7.1)$$

where the a and b parameters are time-varying to account for the change of dynamics as the batch progresses. In order to obtain a combined model the inputs and outputs can be stacked in order to obtains a stacked model. This will be shown in the following.

A new state vector $\mathbf{x}$ is defined. This vector contains all measurements from the entire batch. This assumes equal duration of the batches which is also a common goal in industrial batch processing where it is often attempted to follow a fixed recipe. Thus the new output vector is:

$$\mathbf{x} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad (7.2)$$

where y_i is $y(t = T_s i) = y(k = i)$. The sample time T_s is assumed constant. Past outputs as well as inputs are used to estimate new outputs:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}, \quad (7.3)$$

where $\mathbf{A}$ and $\mathbf{B}$ are the model matrices that define the dynamics of the system and the data vectors are defined by $\mathbf{x}_k = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^\top$, $\mathbf{x}_{k-1} = \begin{bmatrix} y_0 & y_1 & \cdots & y_{n-1} \end{bmatrix}^\top$ and $\mathbf{u}_{k-1} = \begin{bmatrix} u_0 & u_1 & \cdots & u_{n-1} \end{bmatrix}^\top$.

The presented model structure started is an ARX type of model with one $\mathbf{A}$ matrix and hence can be viewed as a state space model similar to the global operator form realisation presented by [Dewilde and van der Veen, 1998].

7.3.1 Estimation

The model parameters must be estimated based on collected input/output data. In order to obtain a parsimonious, physically realistic model causality and model order is introduced into the model.

7.3.1.1 Causality Constraints

In the defined model in (7.3) the model matrices are full and hence the defined model is not causal. If the model is to be used for control it is advantageous if the model is causal.

For state space models of the type $\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}$ some constraints can be imposed if a causal model is desired. In order to ensure a causal model only non-zero elements can be permitted on and below the diagonal in both model matrices.

To obtain a causal model equation (7.3) is rewritten with explicit elements where the new causal structure of the model is introduced

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & \mathbf{0} \\ a_{21} & a_{22} & \vdots \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_{11} & \dots & \mathbf{0} \\ b_{21} & b_{22} & \vdots \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix}. \tag{7.4}$$

The challenge is to estimate the non-zero elements in the matrices $\mathbf{A}$ and $\mathbf{B}$ while maintaining the value zero where intended due to causality. The model (7.4) can be rewritten into

$$x_1 = a_{11}x_0 + b_{11}u_0 \tag{7.5}$$
$$x_2 = a_{21}x_0 + a_{22}x_1 + b_{21}u_0 + b_{22}u_1 \tag{7.6}$$
$$x_3 = a_{31}x_0 + a_{32}x_1 + a_{33}x_2 + b_{31}u_0 + b_{32}u_1 + b_{33}u_2, \tag{7.7}$$

where it is immediately clear that outputs are only affected by past outputs and inputs. These equations result in the following equation system where the parameters to be estimated are extracted as a vector

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_0 & & & & & & u_0 & & & & & \\ & x_1 & & x_0 & & & & u_1 & & u_0 & & \\ & & x_2 & & x_1 & x_0 & & & u_2 & & u_1 & u_0 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{22} \\ a_{33} \\ a_{21} \\ a_{32} \\ a_{33} \\ b_{11} \\ b_{22} \\ b_{33} \\ b_{21} \\ b_{32} \\ b_{33} \end{bmatrix} \tag{7.8}$$

$$\mathbf{x}_3 = \mathbf{F}\boldsymbol{\theta}, \tag{7.9}$$

where $\mathbf{F}$ contains the measured input and output data. Note that $\mathbf{F}$ in general will be sparse and a suitable sparse solver must be chosen to obtain the parameters in an efficient manner.

7.3.1.2 Model Order

For simple ARX models the notion of model order describes the number of past sampled values that should be included in the model. The model order is

described by the values n_a and n_b introduced in equation (7.1). The optimal model order is determined by the sample time, the dynamics of the system, the frequency spectrum of the input signal etc. In reality the system order will also be affected by the signal to noise ratio of the data and the number of samples and batches available. In practice n_a and n_b can vary during the batch. However here they are assumed constant.

For the stacked state space models the order of the ARX model can be transformed into having a model where only the diagonal and a few sub-diagonals are non-zero. The number of non-zero diagonals and sub-diagonals is the order of the model. If a low model order is used the number of parameters can be greatly reduced.

Regularisation methods should be used to further reduce the variance of the model parameters. Model estimation is further described in [Gregersen and Jørgensen, 2002].

7.3.2 Simulation

When the model parameters have been identified the model can be used for simulation of the system. One may use the recursion defined in equation (7.3), but the recursive solution method can be avoided by rewriting the system into a form that leads to a closed form solution.

The ARX model is written

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1}. \tag{7.10}$$

x_0 and $\mathbf{u}_{k-1}$ are assumed known. This is introduced into equation (7.10)

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{c}, \tag{7.11}$$

where $\mathbf{c} = \mathbf{B}\mathbf{u}_{k-1}$. The matrices and vectors are expanded to show their elements for a small example with $n = 4$ and $n_a = 2$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ 0 & a_{32} & a_{33} & \\ 0 & 0 & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}, \tag{7.12}$$

which is simplified into

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} a_{11}x_0 \\ a_{21}x_0 + a_{22}x_1 \\ a_{32}x_1 + a_{33}x_2 \\ a_{43}x_2 + a_{44}x_3 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}. \tag{7.13}$$

It is now the goal to extract an $\mathbf{x}$ vector on the left hand side that is the same as the one found on the right hand side. This is possible by moving the terms

involving x_0 to the $\mathbf{c}$ vector.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \dots & \dots & \mathbf{0} \\ a_{22} & \ddots & \ddots & \vdots \\ a_{32} & a_{33} & \ddots & \vdots \\ \mathbf{0} & a_{43} & a_{44} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} c_1 + a_{11}x_0 \\ c_2 + a_{21}x_0 \\ c_3 \\ c_4 \end{bmatrix} \tag{7.14}$$

$$\mathbf{x} = \tilde{\mathbf{A}}\mathbf{x} + \tilde{\mathbf{c}} \tag{7.15}$$

The following equation system is then formed, which can be used for the calculation of $\mathbf{x}$

$$(\mathbf{I} - \tilde{\mathbf{A}})\mathbf{x} = \tilde{\mathbf{c}}. \tag{7.16}$$

Note that the vector $\tilde{\mathbf{c}}$ depends only on initial state x_0 and the input $\mathbf{u}$. This means that we now have a very simple linear model for simulating a batch process. For analysis purposes it is convenient to give the explicit result for $\mathbf{x}$ as

$$\begin{aligned} \mathbf{x} &= \tilde{\mathbf{L}}^{-1}(\mathbf{A}_0 x_0 + \mathbf{B}\mathbf{u}) \\ &= \tilde{\mathbf{L}}^{-1} \begin{bmatrix} \mathbf{A}_0 & \mathbf{B} \end{bmatrix} \begin{bmatrix} x_0 \\ \mathbf{u} \end{bmatrix} \\ &= \mathbf{G} \begin{bmatrix} x_0 \\ \mathbf{u} \end{bmatrix} \\ &= \mathbf{G}\tilde{\mathbf{u}} \end{aligned} \tag{7.17}$$

where $\mathbf{G}$ is the matrix operator that maps known initial conditions and inputs contained in $\tilde{\mathbf{u}}$ to the outputs.

The derivation of the stacked state space models sofar has been for SISO systems only. It is straightforward to extend the modelling scheme to MIMO system merely by stacking the additional input and output on top of each other.

7.4 Conceptual control of batch processes

When batch processes are operated using a predetermined recipe it is important that the recipe is optimal and robust. Optimisation of processes is in indutry often based on multiple experiments to find optimal initial conditions and optimal trajectories for the outputs and hence the inputs.

Optimisation and control of a batch process can be a multidisciplinary exercise. For the entire plant issues such as planning, down time, fixed and variable costs are very important factors. Here the focus is on the parts of the process that has been modelled using dynamic models.

A simple objective for the optimisation of batch process is to maximise the yield $J(T)$ of the process, where T is the final time of the batch. This simple objective does not account for the price of initial material, down time cost and operating cost. Furthermore, the objective does not account for down

stream processing that may not be performed optimally when the fermentation objective is treated *separately* from the rest of the process.

For the fed-batch example that is dealt with in section 7.6 the yield is defined as $J(T) = X(T)V(T)$, where $X(t)$ is the biomass concentration and $V(t)$ is the volume. The final time is assumed known and fixed, which is the attempted operation strategy although it is not always followed in practice due to disturbances in the process or production line.

Using the causal batch model the optimisation problem can be solved using Quadratic Programming (QP). The optimisation problem can be expressed as

$$\max \mathcal{X}^\top \mathbf{Q} \mathcal{X} \tag{7.18}$$

where $\mathcal{X}$ is the state vector. $\mathbf{Q}$ is a weighting matrix chosen such that $\mathcal{X}^\top \mathbf{Q} \mathcal{X}$ equals $X(T)V(T)$. For this simple object function it is seen that

$$\mathcal{X}^\top \mathbf{Q} \mathcal{X} = \begin{bmatrix} S(0) \\ \vdots \\ S(T) \\ X(0) \\ \vdots \\ X(T) \\ V(0) \\ \vdots \\ V(T) \end{bmatrix}^\top \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 & \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots \\ \vdots & \ddots & \vdots & \ddots & \vdots & 0.5 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & \cdots & 0 & \\ \vdots & \ddots & \vdots & 0.5 & \vdots & \ddots \end{bmatrix} \begin{bmatrix} S(0) \\ \vdots \\ S(T) \\ X(0) \\ \vdots \\ X(T) \\ V(0) \\ \vdots \\ V(T) \end{bmatrix} =$$

$$X(T)V(T) = J(T) \tag{7.19}$$

The optimisation problem (7.18) is further complicated by the constraints imposed by limitations in the actuators and by state constraints. These constraints lead to the following QP problem

$$\begin{aligned} &\max_{\tilde{\mathbf{u}}} \mathcal{X}^\top \mathbf{Q} \mathcal{X} \\ &\text{s.t.} \\ &\mathcal{X} = \bar{\mathbf{x}} + \mathbf{x} \\ &\mathbf{x} = \mathbf{G}\tilde{\mathbf{u}} \\ &\tilde{u}_l(i) \leq \tilde{u}_i \leq \tilde{u}_u(i), \end{aligned} \tag{7.20}$$

where $\mathbf{G}$ and $\tilde{\mathbf{u}}$ were defined in equation (7.17). This problem can be reduced to a standard QP problem

$$\begin{aligned} &\max_{\tilde{\mathbf{u}}} \tilde{\mathbf{u}}^\top \tilde{\mathbf{Q}} \tilde{\mathbf{u}}/2 + \tilde{\mathbf{R}} \tilde{\mathbf{u}} \\ &\text{s.t.} \\ &\tilde{u}_l(i) \leq \tilde{u}_i \leq \tilde{u}_u(i), \end{aligned} \tag{7.21}$$

where $\tilde{\mathbf{Q}} = \mathbf{G}^\top \mathbf{Q}\mathbf{G}$ and $\tilde{\mathbf{R}} = \bar{\mathbf{x}}^\top \mathbf{Q}\mathbf{G}$.

Optimising control is an on-line version of the previously derived optimisation problem. At sample time k the current state is estimated based on the previous $k-1$ measurements of the batch and the model. By using this estimate an optimal input trajectory can be calculated for the part of the model that concerns the remaining $N-k$ samples of the batch. Since the problem also in this case is a QP problem the calculation time will be very small and is directly suitable for on-line implementation.

7.5 Model Predictive Control

A batch process experiences large changes in concentrations, mass balances and time constants as the batch progresses. For a biological process many reactions take place simultaneously. A set of these reactions must be controlled in order to achieve the production goal. This goal is to provide the desired product quantity and quality while limiting the formation of unwanted byproducts. The design of a batch recipe for a fermentation process has the purpose to obtain a high yield while limiting unwanted byproduct formation that will lead to a loss of substrate or may produce toxins. The prevailing operating strategy in today's batch processing plants is to operate the batch processes using recipes. The control system has then the goal to keep the states or measurable outputs as close as possible to the predetermined trajectories.

7.5.1 Optimisation based formulation of the control problem

The Model Predictive Control (MPC) problem is specified by reference trajectories for the outputs and inputs which are defined as vectors $\mathbf{x}^0$ and $\mathbf{u}^0$. A model of the system must also be available. In this paper it will be assumed that the model is a stacked state space model, but the model may originate from a nonlinear state space model if such a model is available. The optimisation problem can be posed as a minimisation (QP) problem

$$\begin{aligned} \min_{\tilde{\mathbf{u}}} \quad & \left\{(\mathbf{x}^0 - \mathbf{x})^\top \mathbf{R}_x (\mathbf{x}^0 - \mathbf{x}) + (\mathbf{u}^0 - \mathbf{u})^\top \mathbf{R}_u (\mathbf{u}^0 - \mathbf{u})\right\} \\ & \text{s.t.} \\ & \mathbf{x} = \mathbf{G}\tilde{\mathbf{u}} \\ & \tilde{u}_l(i) \le \tilde{u}(i) \le \tilde{u}_u(i), \end{aligned} \tag{7.22}$$

where $\mathbf{R}_x$ and $\mathbf{R}_u$ are suitably chosen positive definite weighting matrices. The output reference trajectory is used to drive the process close to the desired performance in order to achieve the desired product quality consistently.

For real problems the input reference signal is not known. It is important to acknowledge that if good control is desired the input signal must not be

constrained too severely. When the sample time is long as it is the case for most biological batch processes large changes in u from sample to sample must be allowed. For batches with a short sample time a penalty on the derivative of u may be included to obtain a smooth u profile. Note that this can be done without change of the MPC problem as stated in equation (7.22) simply by including terms above and below the diagonal of $\mathbf{R}_u$.

7.5.2 Analysis of the Batch Control Problem

The above mentioned MPC control problem is very flexible in the sense that the weighting matrices can be used to select the relative importance of the variables throughout the duration of the batch. However the MPC formulation provides little insight into the control problem. Although the QP problem can be easily solved the solution gives little information about how many outputs that can be controlled simultaneously. Since the system locally is time variant it can easily be the case that some outputs are controllable at some, but not at other points in time. The following presentation of an alternative controller will shed some light on this.

The most simple controller than can be set up is the proportional controller. Such a controller will compensate for or eliminate deviations from the reference trajectory. In the following the derivation of a controller will be shown based on the assumption that a measurement has been obtained at $t = 0$, a suitable model has been identified using data from previous batches and that control is desired of one or more outputs.

The measurements obtained at $t = 0$ is denoted $\mathbf{x}_0$. Using this measurement an input trajectory is calculated

$$\mathbf{u}_1 = \mathbf{K}\mathbf{x}_0, \tag{7.23}$$

where $\mathbf{u}_1$ is the trajectory of inputs form $t = 1$ to $t = T$ and $\mathbf{K}$ $((N-1) \times n_s)$ is the gain matrix of the proportional controller. n_s is the number of states. The problem is to calculate the matrix $\mathbf{K}$, i.e. to tune the controller such that the outputs do not deviate from the reference trajectory $\mathbf{x}^0$. In the following a procedure is given. The measurement trajectory predicted from the control using the model is given by

$$\mathbf{x}_1 = \mathbf{G}\begin{bmatrix}\mathbf{x}_0 \\ \mathbf{u}_1\end{bmatrix}, \tag{7.24}$$

where $\mathbf{G}$ is the "transfer matrix" derived in equation (7.17). By combination of equation (7.23) and (7.24) the disturbance rejection requirement leads to the equation system

$$\mathbf{x}^0 = \mathbf{G}\begin{bmatrix}\mathbf{x}_0 \\ \mathbf{K}\mathbf{x}_0\end{bmatrix} \tag{7.25}$$

By manipulation of the equation above an equation system with the unknown matrix $\mathbf{K}$ (as $\mathrm{vec}(\mathbf{K}^\top)$) can be obtained.

$$\mathbf{x}_i^0 = \mathrm{vec}\left[G_i \otimes \mathbf{x}_0\right]^\top \mathrm{vec}(\mathbf{K}^\top) \quad i = 1, \ldots, (N-1) \cdot n_s, \tag{7.26}$$

where $\otimes$ denotes the Kronecker product and vec$(\cdot)$ is the column vector operator. The matrix made from the rows vec $[G_i \otimes \mathbf{x}_0]^\top, i = 1, \ldots, (N-1) \cdot n_s$ can not be assumed to have full rank and the equation system in (7.26) will have to be solved using a stable method such as the pseudo inverse. The rank of the matrix can be used to assess the number of controllable outputs (through time). Once $\mathbf{K}$ has been obtained it is straightforward to calculate $\mathbf{u}$ using equation (7.23). In order to obtain a smooth input trajectory it is beneficial to regularise $\mathbf{u}$ using Tikhonov regularisation with a first order derivative approximation matrix as regularisation matrix [Hansen, 1996].

Once the control trajectory for the entire batch has been calculated one must decide how many of these input moves that will be implemented before a new update of the control will be performed. This decision must be based upon

- The dynamics of the system.
- The quality of the model.
- The size of the numerical problem and the resulting computational time.
- The stability of the control algorithm.

The last item must of course be analysed further for the control algorithm in general and for the particular system that is to be controlled.

Since the system is controlled by a linear model it is foreseen that if the system enters highly nonlinear operating regions this would result in a degradation of controller performance. The controller may not be able to stabilise the system in this case and large deviations from the desired trajectories may be the result. The traditional notion of stability for time invariant and time-varying linear systems is a valuable analysis tool for these systems [Rugh, 1996; Dewilde and van der Veen, 1998]. Such stability analyses are not relevant for intrabatch batch control since the terminal time T is finite and the inputs for a real system are bounded.

In the proposed algorithms for the calculation of $\mathbf{u}$ it is only possible to smoothen the control signal between control signal updates. Whenever the control trajectory is recalculated there may be a jump in the control signal. Such a jump may be desired or undesired depending on the dynamics of the system and the cost of a change in the control signal. For batch systems where the control signal is only updated rarely a jump can be tolerated, but for short update period some smoothing is necessary. Again the amount of smoothing will depend on the dynamics of the system and the tolerated performance loss which will be the result of the smoothing. It is simply proposed that the new control signal $u^s_{t,new}(t)$ will be a weighted average between the newly calculated control signal $u^s_t(t)$ and the previously implemented control signal $u^s_{t-1}(t)$.

$$u^s_{t,new}(t) = (1 - c_f)u^s_t(t) + c_f u^s_{t-1}(t). \tag{7.27}$$

This first order filter has a tuning time constant c_f. For slow dynamic systems with few disturbances this constant will be close to 1. For systems where

fast response to disturbances must be made the constant c_f must be closer to 0. It is likely that a dynamic scheme can be deviced where c_f may be adjusted according to the size of the deviation from the desired trajectory and the variance of the output estimates.

The proposed control algorithm is shown in algorithm 2.

7.6 Batch Control Example

In section a small example is presented that gives a proof of concept for the developed control and analysis methods presented.

For a fermentation process we have a simple model that can be used for simulation of the system in fed-batch operation. The model contains only three states: substrate concentration S [g/l], biomass concentration X [g/l] and volume of the fermentor V [l]

$$\begin{aligned}\dot{S} &= -\frac{\mu}{0.5}X + (S_f - S)\frac{F}{V}\\ \dot{X} &= \mu X - X\frac{F}{V}\\ \dot{V} &= F,\end{aligned}$$

where the growth rate is given by $\mu = \frac{\mu_{max}S}{K+S+0.5S^2}$. The growth rate depends heavily on S. The maximum growth rate is obtained for $S = 0.245$ g/l. For substrate concentrations larger than this value the growth rate slowly decreases.

As an illustrative example data from 10 simulated batches are used for the estimation of a stacked state space model. In this example the CGLS numerical method is used for the estimation of the model matrices [Hansen, 1998].

Algorithm 2: Algorithm for calculating the control signal for the batch control problem.

(1) Estimate model matrices based on available relevant data.
(2) set $t = 0$.
(3) Obtain the measurements x_t.
(4) Calculate the controller $\mathbf{K}$
(5) Calculate the (smoothed) control signal $\mathbf{u}$ for remaining part of the batch based on the model.
(6) **if** $t > 0$
(7) Let $u^s_{t,new}(t) = (1 - c_f)u^s_t(t) + c_f u^s_{t-1}(t)$.
(8) Implement the control signal for one sample.
(9) **if** Batch is *not* finished
(10) **goto** 3.

The CGLS algorithm is a recursive numerical method for solving sparse least squares problem that regularises the result. As the CGLS algorithm iterates a closer approximation to the least squares solution is obtained with decreasing regularisation of the solution vector [Hansen, 1996]. Table 7.1 shows the number of CGLS iterations used. The optimal number of iterations is determined using cross validation.

For the control problem reference trajectories (setpoint curves) are set up for the three outputs S, X and V. The trajectories are based on a feed flow rate trajectory selected for this problem. This input trajectory is shown in figures 7.3 and 7.4 as dotted lines.

The controller $\mathbf{K}(t=0)$ is calculated from the model using equation (7.26). The matrix to be inverted does not have full rank. The rank for this example is equal to N. This can be seen in figure 7.2 where the singular values for the equation system are shown. A simple interpretation is that it is only possible to control one output by the single input to this system. A full interpretation is that for a truly linear system it is possible to control individual output values at N points in time with one unbounded input. For a bounded input we can only expect to obtain (near perfect) control on a lower number of outputs.

In order to achieve a suitable control scheme we choose to control the biomass concentration and the substrate concentration simultaneously. Thus, for this example it is selected that the volume is allowed to deviate from the reference trajectory freely. The volume and the control signal (the feed flow rate) are closely related and therefore this freedom has to be introduced to allow the input to control S and X. As a result of the analysis performed above it is not expected that the control is perfect, but rather that a tradeoff is achieved between the control of the biomass concentration and the substrate concentration.

The problem simulated is the case where the initial values for the states are 10% higher than in the nominal case (which was used for the model estimation). There is no noise on the system and therefore it is only this *initial* disturbance that has to be rejected by the control algorithm. Since the noise content is low it may be required that the control signal is smooth. Thus a smoothing parameter $c_f = 0.95$ is selected. The solution for this particular problem is very sensitive to changes in c_f. The controller $\mathbf{K}$ is updated at every sample instant. Thus, the newly calculated control signal is implemented for only one sample. Figures 7.3 and 7.4 shows the progress of the batch after the control algorithm has been implemented for 2 hours and the biomass and and substrate concentrations are at this time close to the reference values. The figures also show the calculated control signal and the expected behaviour of the states if this control signal would be implemented unchanged for the rest of the batch. However, the control signal is still updated for each sample for the remanining part of the batch. It is seen that the control signal initially is used to adjust the biomass concentration such that it gets closer to the perscribed trajectory and good accordance is achieved within three quaters of an hour. It can be seen

that the control signal is smooth for the implemented two hours, but oscillatory for the last part of the batch. This oscillation becomes even more pronounced during the very last part of the batch. For this part the control signal has little effect due to the increased volume of the fermentor and the resulting slower dynamics of the system. The reason for the oscillatory control signal towards the end of the batch is probably due to the perfect control requirements since we demand that the deviation from the reference trajectory should be exactly zero. Furthermore, the decreasing accuracy of the model towards the end of the batch also has an effect. Since the volume is increasing as the batch progresses the dynamics becomes slower and hence the systems requires more excitation of the control signal in order to provide information about the dynamics of the system for this last part. Due to the oscillatory behaviour of the control signal the controller and control trajectory are not updated for the last 7 samples of the batch.

The control scheme performs well for this example. The sum of squares of the difference between the actually realised outputs and the reference trajectory is shown in table 7.1. Remember that only the biomass concentration and substrate concentration are controlled and it is therefore expected that the sum of squares for the volume is large—and that it increases as the control

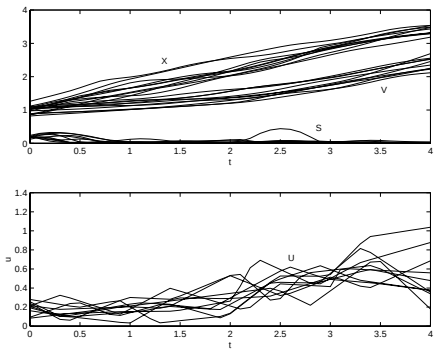


Figure 7.1. Data used for estimating the model for the control example.

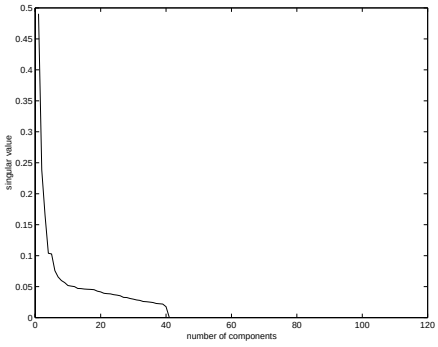


Figure 7.2. Singular values for the matrix inversion in equation (7.26) for the calculation of $\mathbf{K}$.

Batch number	CGLS iterations	ssq_S	ssq_X	ssq_V
1	7	0.2002	0.1307	0.5419
2	20	0.1797	0.1284	0.6695
3	18	0.1527	0.1371	1.2120

Table 7.1. Sum of squares of the difference between the actually realised outputs and the reference trajectory for the controlled batches. The substrate concentration S and biomass concentration X are controlled.

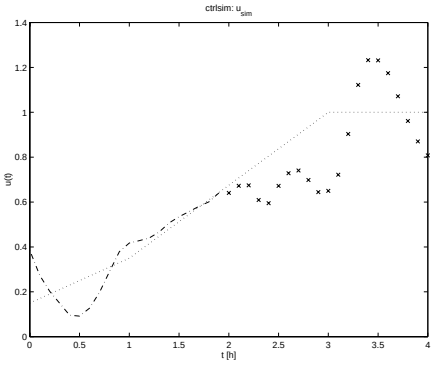


Figure 7.3. Input signal after the first 20 samples for the first controlled batch corresponding to $t = 2$ h. The dotted line shows the reference input trajectory for the nominal case. This trajectory was *not* known to the control algorithm. The dash-dotted line shows the implemented control signal. The crosses show the calculated future control inputs at this point in time.

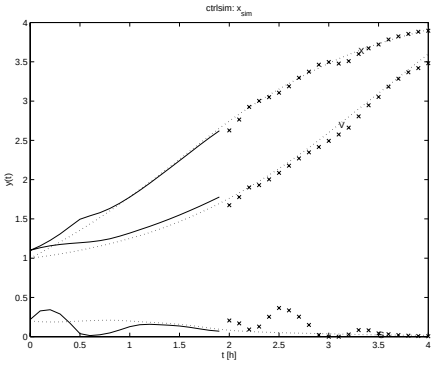


Figure 7.4. Output signal after the first 20 samples for the first controlled batch corresponding to $t = 2$ h. The dotted lines shows the reference trajectories. In this control problem only the biomass (X) and substrate (S) are controlled. The full lines show the realised outputs and the crosses show the expected outputs for the rest of the batch.

performance increases.

After the batch is completed a new set of data has been obtained that can be used for modelling since this new batch is very close to the future intended operation of the batches and therefore is a good candidate dataset to improve the model. The model is updated by including the new data set with data from the previous batches. The data sets are given equal weight in the model. The CGLS method is used to calculate the model matrices and prior to each calculation the optimal number of CGLS iterations is found based on a cross validation scheme using the previously obtained data for validation. The number of iterations are shown in table 7.1.

The modelling and control iteration is performed over three batches in order to evaluate the control performance improvement. The initial conditions have been set to be the *same* for all three batches at 10% above the nominal values. The sum of squares for the biomass concentration and the substrate concentration do decrease as shown in table 7.1. Since this decrease requires control of the feed flow rate the volume *must* deviate more from the reference trajectory than the other two states.

The controller performance can be further assessed by inspecting figures 7.5 and 7.6 that show the input and output signals respectively for the entire third batch in table 7.1. It is seen that the deviations are small throughout the batch and that the initial deviation is smoothly compensated for. The control signal is for this example rather smooth. This is due to the large value of c_f.

The control scheme, although very simple, does work for the example case given. The limitations of using a linear model do appear if the reference trajectory of the biomass is increased. Then the control signal will have larger

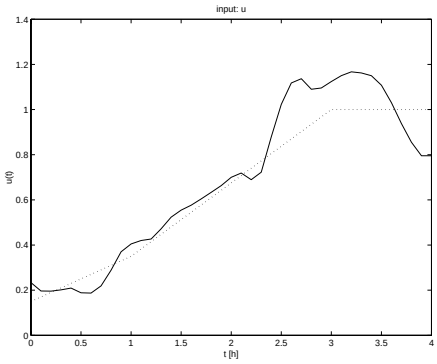


Figure 7.5. Input signal for the third controlled batch. The full line show the realised input signal whereas the dotted line show the reference input signal for the nominal case.

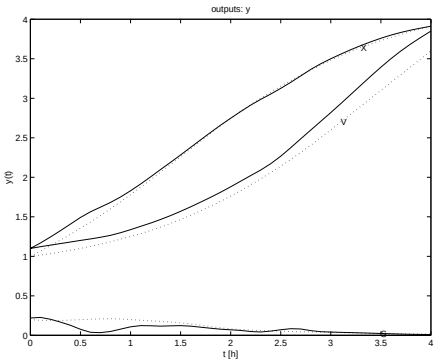


Figure 7.6. Output signal for the third controlled batch. The dotted lines show the reference trajectories and the full lines show the realised trajectories for the entire batch.

fluctuations sometimes yielding a large substrate concentration which induces a sign change of the system. This sign change cannot be reflected in the model and therefore the controller will take wrong actions. There seems to be no way of handling this control deficiency when using only a single linear model.

The control method has a large set of decisions and tuning parameters.

- When updating the model how much weight should be placed on the various data sets? In addition a forgetting factor could be included.
- How many CGLS iterations should be performed?
- What outputs should be controlled and when?
- What sample interval should be used?
- How often should the control trajectory be updated?
- c_f should be selected to accommodate the dynamics of the system.

7.7 Discussion and Conclusions

Linear dynamic models has been used for a long time in process control and system analysis. There are many reasons for the the large focus on linear models. Some reasons are the ease with which linear models may be used for systems analysis, control design and implementation in real-time systems and also the ease with which linear models may be obtained either from process data or derived from process knowledge. The model type used in this paper is called stacked state space models.

This paper has shown how batch process optimisation and batch process control problems may be formulated using the developed stacked state-space models.

The process models are extensions of existing linear models and therefore easily integrated into existing algorithms for process optimisation and process control although the process models are much larger than normally seen for continuous systems and special attention should be given to the finite duration of a batch process. The latter requires special handling of initial conditions, which often have a severe influence on the course of the batch. Terminal conditions and the assumed known duration of the batch also introduce constraints in the control and optimisation problems.

The example used in this paper is based on a fed-batch bioreactor case. Many other examples of batch processes exist in chemical plants and elsewhere and the methods presented here can be used for these batch processes as well. In addition one may apply the methods presented here to un-steady state operation in continuous processes such as grade changes.

The MPC formulation for control of the batch is straightforward and only special attention should be given to the formulation of the weight matrices. This is a tuning problem. The proportional controller introduced here can be a useful tool for analysis of the control problem, but it is expected that an application will utilise a full MPC implementation using carefully selected weighting matrices, sample time and update interval.

In this paper the discussion about where the model should come from has been omitted. However, it is avantageous for the method to know that the model may be derived either from data or from process knowledge in form of a first principles model. This fact will make it easier to implement a controller based on these methods using first a model based on process data and subsequently, if necessary, an improved model based on first principles, where such knowledge can be obtained and will offer additional accuracy over the identified model.

Using linear models for process optimisation and control offer a potential problem if the system under consideration is nonlinear. The linear model is only approximately accurate locally and the linear model may show severe system/model mismatch if the system deviates too much form the trajectories where the model has been derived. This may especially be a problem when optimising since such operations are meant to change the trajectories to obtain the desired goals and performance. One way of addressing these problem is to use regression trees or other nonlinear models in the modelling [de Veaux *et al.*, 1993].

Extensions to batch control problems do exist that investigates the interbatch control problem [Lee *et al.*, 1997]. This is a topic of increasing interest and is the subject of further investigations as the model type presented here is also directly applicable for iterative learning control.

8

Discussion and conclusions

This chapter will summarise the main results of this thesis and present ideas for future work in this area. The work is divided into two large areas: Process chemometrics for fault diagnosis and stacked state space models.

First a summary of the fault diagnosis using chemometric models and the stacked state space models will be given in sections 8.1 and 8.2, respectively. In section 8.3 the main results of this work will be pointed out followed by a description of proposals for future work in section 8.4.

The goal of this thesis is to investigate methods for developing models based on process data from industrial scale fermentation processes. The availability of prior knowledge has been assumed very limited. Especially the use of first principles modelling, which potentially could have improved the modelling immensely have been assumed unavailable.

The models developed in this thesis are all linear although it was not set as an initial goal for the modelling. The linear model structure makes it possible to define the structure of the models using simple matrix notation. The model parameters are easily estimated using stndard chemometric methods and other numerical methods. The estimation problems and the application of the models for fault diagnosis, system simulation, optimisation and control are computationally fast and should therefore allow the modeller to perform explorative data analysis in an highly interactive manner. Since the calculations are fast and can be performed interactively it is possible for the modeller to develop many model candidates and either select one model candidate for the application or to use many model candidates, e.g. one for each specific application.

8.1 Fault diagnosis

Fault diagnosis consists of

- Detection
- Isolation
- Identification

where fault detection is a matter of detecting if and when a fault has occurred in a system. This is usually done by creating a residual between system be-

haviour and the expected system behaviour based on the available knowledge about the system. In this thesis the knowledge has been assumed to be represented through chemometric models based on process data. When a fault has been detected it is important that the fault can be isolated. The purpose of the isolation task is to isolate the variable or variables that are deviating from normal system behaviour and therefore signal the error. Faults must be compensated for or eliminated. This fault revovery requires that the fault is identified. Fault identification means that the physical origin of the fault must be found and a solution strategy must be outlined through a cause and effect analysis of the system. The fault identification task requires detailed knowledge of the system especially tailored to handle faults. Since it is often not possible to obtain process data under the event of all possible faults the identification task requires external knowledge often in the form of the operator's process knowledge and fault recovery will therefore be handled by manual intervention by the process operator.

The process chemometrics tools mentioned in this thesis will assist the operator in detecting and isolating the faults mainly through the use of graphical displays of process statistics and statistical tests.

The real plant data used for modelling for fault diagnosis have been from a large-scale industrial fermentation fed-batch process producing an enzyme. The process has been running under normal operating conditions. I.e. no special experiments have been performed to obtain the data.

Fault diagnosis may be performed in different ways:

- Using on-line data to detect *any* fault in the operation of the process. The presented method is based on PCA (or MPCA).
- Using on-line data and one or more quality variables to detect faults that effect the quality variables used in the modelling. The method presented is based on PLS (or MPLS).
- Using all available measurements separated into inputs and outputs. The method presented in based on stacked state space models.

For processes where quality variables *are* available it is important to utilise those since the fault diagnosis then becomes more sensitive towards faults that affect the quality. Many processes exist where the quality is not easily quantifiable and one therefore have to use a PCA based method.

Process chemometrics tools have been used to perform the initial analysis of the data using PCA. This type of analysis provides a quick overview of many aspects of the data and although the PCA does not in all cases lead to a usable conclusion it may often show where to look for abnormalities in the data.

It has been demonstrated how to detect faults using a MPLS model developed using real plant data. The quality variable used is the final enzyme concentration (activity). It has been demonstrated that the developed MPLS model can be used for fault detection and isolation using real data for the modelling. Plots have been presented that facilitates these tasks.

A stacked state space model was demonstrated using simulated data to be able to detect faults. The stacked state space model is an input/output model type. By separating the measurements into inputs and outputs and by explicitly utilising the dynamics of the process one can detect changes in the dynamics of the process im addition to detecting changes in the level of the process variables. Creating a model based on on the knowledge of the dynamics of the process may be a powerful addition to the PCA and PLS type models for fault diagnosis, but more examples with real data are necessary to ascertain if the extra work in terms of modelling effort and collection of dynamic data is worth the effort.

8.2 Stacked State Space Models

Batch processes are often described using nonlinear and/or time-varying models often with complicated dynamics and rate expressions, but their behaviour may be approximated with a set of local linear models. A modelling framework suitable for data based process identification has been developed to overcome this modelling challenge using a linear model structure. A model structure similar to an ARX model used for linear, stationary systems has been extended to allow for time-varying processes and the special fixed duration of batch processes.

It has been described how a batch process that is operated near a reference trajetory can be modelled as a set of linear time invariant models. This set of models is combined into a single model where all parameters are estimated simultaneously by stacking time shifted models on top of each other in order to cover the entire batch. This model type is called a stacked state space model.

The linear model structure for the stacked state space model contains many parameters, but by using regularisation and other numerical tools it is assured that numerical stability can be obtained even under noisy conditions. By furthermore fixing parameters at prior know values more reliable models may be identified.

An outline is given on how to incorporate process knowledge in the black box models. Often partial knowledge is available about some of the kinetics expressions or mass balances. This knowledge may improve the model when it is possible to merge this information with the identification scheme. It is equally important to eliminate interactions when the interactions are known not to exist.

Problem formulations are given for the batch optimisation problem and the batch control problem. These methods may be used concurrently on a batch online to control and optimise simultaneously.

A proportional control formulation of the control problem is stated and solved using an example where it is shown that by incorporating data from new batches a better model and hence an improved control performance can be obtained.

8.3 Main results

This thesis collects ideas and formulas for data mining, process chemometrics and process control. Although it is difficult to give recipes for data mining or modelling general processes an attempt have been given outlining the important issues of repeated data validation and pre-treatment of the data prior to the modelling phase.

Few attempts have been made at using process chemometrics for fault diagnosis on biotechnological processes with standard (cheap) measurements. This thesis shows some of the many problems that lie in maintaining the quality of these data, but it also demonstrates the potential of using processes chemometrics on these type of processes. Prototype software tools have been developed for this work that presents the idea of an integrated, interactive tool for data mining. The developed tools handles the data sets as they are obtained from the process and facilitates data pre-treatment and chemometric modelling. Tools of this type will facilitate the application of process chemometrics in a wider forum than presently observed.

When variables may be separated into a set of input variables (actuators) and output variables (states) a causal input/output model may be formed. A new method for estimating parameters in stacked spate space models utilising regularisation have been set up. This model type is able to capture the time-varying and nonstationary features of the batch process using a linear model structure. Formulations have been given on how this model structure may be used for process control and optimisation that are not new in themselves, but it is a new result that such models may be estimated from batch plant data that are directly suitable to be used for process optimisation and process control applications. A new proportional controller for batch control was designed and demonstrated.

Finally, a comparison between the correlation type chemometric model and the causal batch ARX models was given. These two types of models are similar in type and to some extent in the computations involved. It was shown that by using the causal model in the fault diagnosis framework a separation of faults in the input and the outputs may be obtained.

The stacked state space models have the advantage over the noncausal process chemometrics models that they have fewer parameters to be estimated since the causality constraints reduces the number of interactions in the model. The stacked state space model is thus a subset of the standard chemometric model for process monitoring. In the case where one wants to use the stacked state space model for process monitoring one has to consider the requirements on obtaining process data that describes the dynamics of the process and that the requirements for creating an accurate dynamic model are higher than for creating a standard process chemometic model of PCA or PLS type. In order to identify a dynamic model one has to excite the inputs of the system to capture

the input/output relationship and much more carefully select the model order and sample time in order to achieve a proper model.

8.4 Future work

This thesis has been an exercise in linear modelling. It has been a most interesting experience to see how far one could get using input/output model as opposed to first principles models. Obvious limitations stems from only using linear models. Extension of the stacked state space models to include (slightly) nonlinear terms must be investigated. Related nonlinear model structures may also be cast into the stacked state space model framework. These structures include artificial neural networks; nonlinear PCA and nonlinear PLS; nonlinear time series models and models based partially on process knowledge. An extension of the modelling scheme to model the parameters as splines with suitable chosen nodes seems obvious.

Regularisation methods have been used to obtain stable solutions for the continuous models. It may be possible to smoothen the solution using wavelets in the estimation or to perform the entire estimation in the frequency domain.

The notation in this thesis have been kept in a language suitable for chemometricians and people knowledgeable in linear algebra. It it probably less readable by statisticians and to an even lesser extent to people trained in process control. I believe that using the methods and notation chosen it has become possible to join the worlds of process chemometrics and process control together, but it is also acknowledged that there is a need to rewrite the methods into something that is more aligned with the process control literature.

The linear structure of the models makes it possible to analyse many aspects of the models. Extensions to include experimental design approaches and analysis of process stability could be carried out using the proposed models.

Application of the mentioned methodologies is not limited to fermentation processes, but can be used for other batch, semi-batch and periodic chemical and non chemical processes.

It may be possible to learn from other disciplines that in their application are very different from chemical processes, but share the same estimation problems. Such applications could be image processing or robotics where problems are solved that are similar to the ones seen for chemical batch systems usually with algorithm that are fast and efficient.

A

Linear Algebra

This appendix is by no means a full treatment on the subject of linear algebra, but merely an introduction to the notation and methods used in this thesis. The theorems (which are not presented as such) are given without proof. They can be found in many books related to linear algebra [e.g., Golub and van Loan, 1991].

A.1 Vector and Matrix Notation

Symbols for vectors and matrices will be put in bold, e.g. $\mathbf{v}$, $\mathbf{X}$. Scalers will be put in italic: e.g. x.

Variables denoting vectors will in this thesis always be column vectors. A row vector will be denoted $\mathbf{v}^\top$.

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_I \end{bmatrix} \qquad \mathbf{v}^\top = \begin{bmatrix} v_1 & v_2 & \cdots & v_I \end{bmatrix}. \tag{A.1}$$

A matrix $\mathbf{X}$ is an array of numbers x_{ij} ordered in I rows and J columns. We say it has dimension $I \times J$ and we can either write $\mathbf{X}$ $(I \times J)$ or $\mathbf{X} \in \mathbb{R}^{I \times J}$. When we look at the individual elements of $\mathbf{X}$ we can write

$$\mathbf{X} = (x_{ij}) \tag{A.2}$$

for short or use the larger description

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1J} \\ x_{21} & x_{22} & \cdots & x_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ x_{I1} & x_{I2} & \cdots & x_{IJ} \end{bmatrix}. \tag{A.3}$$

Sometimes it is useful to have a notation for a single column or row. The jth column of $\mathbf{X}$ is denoted $\mathbf{x}_j$. The ith row is denoted $\mathbf{x}_{(i)}^\top$. Notice that $\mathbf{x}_{(j)}$ is the ith *row* written as a *column*.

We have thus four ways of writing the matrix $\mathbf{X}$

$$\mathbf{X} = (x_{ij}) = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_J \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{(1)} \\ \mathbf{x}_{(2)} \\ \vdots \\ \mathbf{x}_{(I)} \end{bmatrix}. \tag{A.4}$$

A.1.1 Kronecker product

The Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1J} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2J} \\ \vdots & \vdots & & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \dots & a_{IJ} \end{bmatrix} \tag{A.5}$$

A.2 Rank

The rank of a matrix $\mathbf{A}$ $(I \times J)$ is defined by

$$\text{rank}(\mathbf{A}) = \dim(\text{span}(\mathbf{A})). \tag{A.6}$$

The matrix is said to have *full row rank* if $I \leq J$ and $\text{rank}(\mathbf{A}) = I$. Dually it is said to have *full column rank* if $J \leq I$ and $\text{rank}(\mathbf{A}) = J$.

A.3 Eigenvalues and Eigenvectors

Consider the equation

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}, \tag{A.7}$$

where $\mathbf{A}$ $(J \times J)$ is a square matrix. Solution vectors $\mathbf{v}$ that are not equal to the zero vector are called eigenvectors of $\mathbf{A}$. The corresponding λ-values are called eigenvalues.

If the matrix $\mathbf{A}$ is symmetric all the eigenvalues are real. If $\mathbf{A}$ is positive (semi)definite all the eigenvalues are greater than (or equal) to zero. Without loss of generality the eigenvalues can be ordered in a non-increasing order.

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_J \geq 0 \tag{A.8}$$

A symmetric matrix with distinct eigenvalues $\mathbf{A}$ can be written as

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top, \tag{A.9}$$

where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_J)$ and the columns of $\mathbf{U}$ are the eigenvectors of $\mathbf{A}$.

Eigenvalues and eigenvectors are by statisticians called latent roots and latent vectors respectively. In this text the former introduced terms will be used.

The spectral radius of a matrix $\mathbf{A}$ is defined as

$$\rho(\mathbf{A}) = \max_{1 \leq i \leq n} |\lambda_i| \tag{A.10}$$

A.4 Singular Value Decomposition

Singular value decomposition (SVD) is a way to decompose a matrix $\mathbf{A}$ $(I \times J)$ as

$$\mathbf{A} = \mathbf{U}\mathbf{M}\mathbf{V}^\top, \tag{A.11}$$

where $\mathbf{U}$ is $(I \times I)$, $\mathbf{M}$ is $(I \times J)$ and $\mathbf{V}$ is $(J \times J)$.

$\mathbf{U}$ and $\mathbf{V}$ are both orthonormal

$$\mathbf{U}\mathbf{U}^\top = \mathbf{U}^\top\mathbf{U} = \mathbf{I} \tag{A.12}$$

$$\mathbf{V}\mathbf{V}^\top = \mathbf{V}^\top\mathbf{V} = \mathbf{I} \tag{A.13}$$

$\mathbf{M}$ is diagonal and contains the singular values

$$\mathbf{M} = \begin{cases} \begin{bmatrix} \mathbf{M}_1 & \mathbf{0} \end{bmatrix} & \text{if } I < J \\ \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{0} \end{bmatrix} & \text{if } I > J \end{cases} \tag{A.14}$$

where

$$\mathbf{M}_1 = \begin{bmatrix} \sigma_1 & & & \mathbf{0} \\ & \sigma_2 & & \\ & & \ddots & \\ \mathbf{0} & & & \sigma_p \end{bmatrix} \tag{A.15}$$

and

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0, \text{ where } p = \min(I, J). \tag{A.16}$$

If $\mathbf{X}$ has not full rank we have

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0, \tag{A.17}$$

where r is the rank of $\mathbf{X}$.

The largest singular value σ_1 can be determined by the following maximum

$$\sigma_1 = \max_{\mathbf{y} \in \mathbb{R}^I, \mathbf{x} \in \mathbb{R}^J} \frac{\mathbf{y}^\top \mathbf{A} \mathbf{x}}{\|\mathbf{y}\| \, \|\mathbf{x}\|} \tag{A.18}$$

A.4.1 Economy Size SVD

The economy size or reduced SVD can be calculated for $\mathbf{A}$ $(I \times J)$.

The result can now be written $\mathbf{A} = \mathbf{U}\mathbf{M}\mathbf{V}^\top = \tilde{\mathbf{U}}\tilde{\mathbf{M}}\tilde{\mathbf{V}}^\top$, and the matrices are now smaller: $\tilde{\mathbf{U}}$ is $(I \times r)$, $\tilde{\mathbf{M}}$ is $(r \times r)$ and $\tilde{\mathbf{V}}$ is $(J \times r)$. r is the rank of $\mathbf{A}$.

A.5 Norm

The norm of a vector or a matrix is a positive number that is often used to summarise the magnitude of the individual elements in the vector or matrix.

A.5.1 Vector Norm

A vector norm is a function $f : \mathbb{R}^I \mapsto \mathbb{R}$ that satisfies the following properties

$$f(\mathbf{x}) \geq 0, \quad f(\mathbf{x}) = 0 \text{ iff } \mathbf{x} = \mathbf{0} \tag{A.19}$$

$$f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y}) \tag{A.20}$$

$$f(\alpha\mathbf{x}) = |\alpha| f(\mathbf{x}). \tag{A.21}$$

A double bar notation is used instead of using the function name $f : f(\mathbf{x}) = \|\mathbf{x}\|$.

A class of norms that are often used is the p-norms, which are defined by

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{I} |x_i|^p \right)^{1/p}, \quad p \geq 1 \tag{A.22}$$

The cases where $p = 1$, 2 and ∞ are often used. The p is added to the double bar notation as a subscript

$$\begin{array}{lcl} \|\mathbf{x}\|_1 & = & |x_1| + \cdots + |x_I| \\ \|\mathbf{x}\|_2 & = & (x_1^2 + \cdots + x_I^2)^{1/2} = (\mathbf{x}^\top\mathbf{x})^{1/2} \\ \|\mathbf{x}\|_\infty & = & \max |x_i|. \end{array} \tag{A.23}$$

A.5.2 Matrix Norms

A matrix norm can be defined similar to equations (A.19)–(A.21).

$$\|\mathbf{A}\| \geq 0, \quad \|\mathbf{A}\| = 0, \text{ iff } \mathbf{A} = \mathbf{0} \tag{A.24}$$

$$\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\| \tag{A.25}$$

$$\|\alpha\mathbf{A}\| = |\alpha| \|\mathbf{A}\|. \tag{A.26}$$

The p-norms are defined by

$$\|\mathbf{A}\|_p = \sup_{\mathbf{x} \neq \mathbf{0}} \left\| \mathbf{A} \frac{\mathbf{x}}{\|\mathbf{x}\|_p} \right\|_p = \max_{\|\mathbf{x}\|_p = 1} \|\mathbf{A}\mathbf{x}\|_p. \tag{A.27}$$

The 1-norm and ∞-norm is easily calculated

$$\|\mathbf{A}\|_1 = \max_{1 \le j \le J} \sum_{i=1}^{I} |a_{ij}|$$
$$\|\mathbf{A}\|_\infty = \max_{1 \le i \le J} \sum_{j=1}^{J} |a_{ij}|,$$

but the calculation of the 2-norm of a matrix is an iterative procedure. It can be shown that the 2-norm of a matrix is equal to its largest singular value.

The Frobenius norm is equal to

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{I} \sum_{j=1}^{J} |a_{ij}|^2}.$$

It can be calculated using the SVD of $\mathbf{A}$ and is given by

$$\|\mathbf{A}\|_F = \sigma_1^2 + \cdots + \sigma_p^2, \quad p = \min(I, J)$$

A.6 Pseudo Inverse

The pseudo inverse of a matrix $\mathbf{A}$ $(I \times J)$ is defined as

$$\arg \min_{\mathbf{X} \in \mathbb{R}^{J \times I}} \|\mathbf{AX} - \mathbf{I}\|_F,$$

where $|\cdot|_F$ denotes the Frobenius norm, see section A.5. The pseudo inverse will be called $\mathbf{A}^\#$.

It can be shown that the pseudo inverse satisfies the four Moore–Penrose conditions

$$\mathbf{A}\mathbf{A}^\#\mathbf{A} = \mathbf{A} \qquad \mathbf{A}^\#\mathbf{A}\mathbf{A}^\# = \mathbf{A}^\# \tag{A.28}$$
$$(\mathbf{A}\mathbf{A}^\#)^\top = \mathbf{A}\mathbf{A}^\# \qquad (\mathbf{A}^\#\mathbf{A})^\top = \mathbf{A}^\#\mathbf{A}. \tag{A.29}$$

The pseudo inverse is defined by (A.28) and its uniqueness is established by (A.29).

If $\text{rank}(\mathbf{A}) = J$, then $\mathbf{A}^\# = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$, while if $\text{rank}(\mathbf{A}) = I = J$, then $\mathbf{A}^\# = \mathbf{A}^{-1}$.

The pseudo inverse can be calculated using the SVD. We have

$$\mathbf{A} = \mathbf{U}\mathbf{M}\mathbf{V}^\top.$$

Using equations (A.12) and (A.13) we can invert $\mathbf{A}$ and get

$$\mathbf{A}^{-1} = \mathbf{V}\mathbf{M}^{-1}\mathbf{U}^\top, \tag{A.30}$$

Remember that the matrix $\mathbf{M}$ is diagonal and is inverted by simply inverting each diagonal element individually ($1/\sigma_p$). In the case where $\mathbf{A}$ does not have full rank the last diagonal elements of $\mathbf{M}$ are equal to zero and can not be inverted. The inverse in this case is then calculated by substituting $1/\sigma_p$ with 0 if $\sigma_p = 0$ (it is not very often that one gets to set $\infty = 0$).

The calculation of the pseudo inverse using the SVD can be summarised as

$$\mathbf{A}^{\#} = \mathbf{V}\mathbf{M}^{\#}\mathbf{U}^{\top}, \tag{A.31}$$

Where $\mathbf{M}^{\#}$ is defined as

$$\mathbf{M}^{\#} = \text{diag}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_r}, 0, \ldots 0\right). \tag{A.32}$$

A.7 Derivatives

The derivative of $f(\mathbf{X})$ with respect to $\mathbf{X}$ $(I \times J)$ is defined by [Mardia *et al.*, 1995]

$$\frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} = \left(\frac{\partial f(\mathbf{X})}{\partial x_{ij}}\right).$$

We have the more useful expressions:

$$\frac{\partial \mathbf{A}^{\top}\mathbf{x}}{\partial \mathbf{x}} = \mathbf{A}. \tag{A.33}$$

$$\frac{\partial \mathbf{x}^{\top}\mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{x}, \quad \frac{\partial \mathbf{x}^{\top}\mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{A}\mathbf{x}, \quad \frac{\partial \mathbf{x}^{\top}\mathbf{A}\mathbf{y}}{\partial \mathbf{x}} = \mathbf{A}\mathbf{y}. \tag{A.34}$$

A.8 Matrix Exponential Function

We use the matrix exponential function for converting from continuous time to discrete time system matrices. The matrix exponential can be defined as

$$\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \frac{1}{n!}\mathbf{A}^n. \tag{A.35}$$

This equation is not suitable for numeric computations of the matrix exponential function. See [Moler and van Loan, 1978] for multiple ways to calculate the matrix exponential.

The fundamental equations for the time invariant system

$$\dot{\mathbf{z}} = \mathbf{A}_c\mathbf{z}(t) + \mathbf{B}_c\mathbf{v}(t)$$

is given by [Rugh, 1996]

$$\mathbf{A}_d = \exp(\mathbf{A}_c T_s) \tag{A.36}$$

$$\mathbf{B}_d = \int_0^{T_s} \exp(\mathbf{A}_c\tau)\, d\tau\ \mathbf{B}_c. \tag{A.37}$$

This can also be written as [van Loan, 1994]:

$$\begin{bmatrix} \mathbf{A}_d & \mathbf{B}_d \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \exp\left(\begin{bmatrix} \mathbf{A}_c & \mathbf{B}_c \\ \mathbf{0} & \mathbf{0} \end{bmatrix} T_s \right). \tag{A.38}$$

This matrix is easy to work with when both $\mathbf{A}_c$ and $\mathbf{B}_c$ need to be converted. We may look at each block in $\left[\begin{smallmatrix} \mathbf{A}_c & \mathbf{B}_c \\ \mathbf{0} & \mathbf{0} \end{smallmatrix}\right]$ and use the series expansion in (A.35)

$$\begin{aligned} \begin{bmatrix} \mathbf{A}_d & \mathbf{B}_d \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \exp\left(\begin{bmatrix} \mathbf{A}_c T_s & \mathbf{B}_c T_s \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right) = \\ \mathbf{I} + \begin{bmatrix} \mathbf{A}_c T_s & \mathbf{B}_c T_s \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \frac{1}{2!} \begin{bmatrix} \mathbf{A}_c^2 T_s^2 & \mathbf{A}_c \mathbf{B}_c T_s^2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \\ \frac{1}{3!} \begin{bmatrix} \mathbf{A}_c^3 T_s^3 & \mathbf{A}_c^2 \mathbf{B}_c T_s^3 \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \dots \end{aligned} \tag{A.39}$$

It is seen that the blocks resemble the series for the matrix exponential and that is used to convert the above expression back to the exponential functions.

$$\begin{aligned} \begin{bmatrix} \mathbf{A}_d & \mathbf{A}_c \mathbf{B}_d \\ \mathbf{0} & \mathbf{I} \end{bmatrix} &= \begin{bmatrix} \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{A}_c^n T_s^n & \left(\sum_{n=1}^{\infty} \frac{1}{n!} \mathbf{A}_c^n T_s^n \right) \mathbf{B}_c \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \exp(\mathbf{A}_c T_s) & (\exp(\mathbf{A}_c T_s) - \mathbf{I}) \mathbf{B}_c \\ \mathbf{0} & \mathbf{I} \end{bmatrix}. \end{aligned} \tag{A.40}$$

When $\mathbf{A}_c = \log(\mathbf{A}_d)/T_s$ is invertible $\mathbf{B}_d$ can be found directly by

$$\mathbf{B}_d = T_s (\log(\mathbf{A}_d))^{-1} (\mathbf{A}_d - \mathbf{I}) \mathbf{B}_c \tag{A.41}$$

The matrix logarithms is associated with the matrix exponential function as it is the case for the scalar case [Dieci *et al.*, 1996]. Given a matrix $\mathbf{T} \in \mathbb{R}^{n \times n}$, any matrix $\mathbf{X} \in \mathbb{R}^{n \times n}$ such that $\exp(\mathbf{X}) = \mathbf{T}$ is a matrix logarithm of $\mathbf{T}$, and one writes $\mathbf{X} = \log(\mathbf{T})$.

Any invertible matrix has a logarithm (not necessarily real). There exists a real $\mathbf{X} = \log(\mathbf{T})$ if and only if $\mathbf{T}$ has an even number of Jordan blocks of each size for every negative eigenvalue. If $\mathbf{T}$ has any eigenvalue on the negative real axis, then no real logarithm of $\mathbf{T}$ can be a primary matrix function of $\mathbf{T}$.

Series expansions also exists for the matrix logarithm. Only a simple one will be shown here. Let $\mathbf{A} = \mathbf{I} - \mathbf{T}$ and assume $\rho(\mathbf{A}) < 1$

$$\log(\mathbf{T}) = \log(\mathbf{I} - \mathbf{A}) = \sum_{k=1}^{\infty} \frac{1}{k} \mathbf{A}^k. \tag{A.42}$$

Further series expansions and Padé approximations can be found in [Dieci *et al.*, 1996].

B

Statistical Analysis

The first section (B.1) will describe some basic formulas used in statistical data analysis.
The subsequent sections (B.2–B.3) will give a very short description of the proporties of random variables.
In this thesis there is not a notationally difference between random variables and their realisations: random data.

It is assumed that all data are real valued.

Proofs and further information can be found in Mardia *et al.* [1995].

B.1 Summarising Statistics

We look at the data matrix

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1J} \\ x_{21} & x_{22} & \cdots & x_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ x_{I1} & x_{I2} & \cdots & x_{IJ} \end{bmatrix}.$$

When I or J are large it will be difficult to handle the elements of the data matrix individually. Therefore some summary statistics are made which describe the size of the data elements and the relations between them.

B.1.1 Sample Mean

The sample[1] mean of the jth variable is

$$\bar{x}_j = \frac{1}{I} \sum_{i=1}^{I} x_{ij}. \tag{B.1}$$

[1] The term "sample", which denotes that the mean is calculated on basis of sampled data, is omitted whenever possible without causing confusion.

The vector of means

$$\bar{\mathbf{x}} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_J \end{bmatrix} \tag{B.2}$$

is called the sample mean vector.
It then follows that

$$\bar{\mathbf{x}} = \frac{1}{I}\sum_{i=1}^{I} \mathbf{x}_{(i)} = \frac{1}{I}\mathbf{X}^\top \mathbf{1}, \tag{B.3}$$

where $\mathbf{1}$ is a column vector of I ones.

B.1.2 Sample Variance

The sample variance of the jth variable is

$$s_{jj} = s_j^2 = \frac{1}{I}\sum_{i=1}^{I}(x_{ij} - \bar{x}_j)^2. \tag{B.4}$$

The number s_j is called the standard deviation.
The sample covariance between the ith and the jth variables is

$$s_{ij} = \frac{1}{I}\sum_{r=1}^{I}(x_{ri} - \bar{x}_i)(x_{rj} - \bar{x}_j). \tag{B.5}$$

The sample covariance matrix is defined by

$$\mathbf{S} = (s_{ij}) \tag{B.6}$$

The explicit expression for $\mathbf{S}$ using vector and matrix notation is

$$\mathbf{S} = \frac{1}{I}\mathbf{X}^\top\mathbf{X} - \bar{\mathbf{x}}\bar{\mathbf{x}}^\top = \frac{1}{I}\left(\mathbf{X}^\top\mathbf{X} - \frac{1}{I}\mathbf{X}^\top\mathbf{1}\mathbf{1}^\top\mathbf{X}\right)$$

If we define the centring matrix as

$$\mathbf{H} = \mathbf{I} - \frac{1}{I}\mathbf{1}\mathbf{1}^\top, \tag{B.7}$$

we get

$$\mathbf{S} = \frac{1}{I}\mathbf{X}^\top\mathbf{H}\mathbf{X}.$$

Because $\mathbf{H}$ is a symmetric idempotent matrix ($\mathbf{H} = \mathbf{H}^\top$, $\mathbf{H} = \mathbf{H}^2$) it follows that for any vector $\mathbf{a}$

$$\mathbf{a}^\top\mathbf{S}\mathbf{a} = \frac{1}{I}\mathbf{a}^\top\mathbf{X}^\top\mathbf{H}^\top\mathbf{H}\mathbf{X}\mathbf{a} = \frac{1}{I}\mathbf{y}^\top\mathbf{y} \geq 0, \tag{B.8}$$

Where $\mathbf{y} = \mathbf{H}\mathbf{X}\mathbf{a}$. It has thus been shown that the covariance matrix $\mathbf{S}$ is positive semi-definite ($\mathbf{S} \geq 0$).

B.1.2.1 Unbiased Estimate of the Variance

The expression for the covariance matrix used so far is the maximum likelihood estimator of the population covariance matrix. Usually another matrix is used. It is defined by

$$\mathbf{S}_u = \frac{I}{I-1}\mathbf{S}. \tag{B.9}$$

The quantity $I-1$ is called the degrees of freedom and is one less than I because the data has been used to estimate $\bar{\mathbf{x}}$. The reason for using $\mathbf{S}_u$ is that it gives an unbiased estimate of the population covariance matrix. That is:

$$E(\mathbf{S}_u) = \mathbf{\Sigma}$$

The estimates $\mathbf{S}$ and $\mathbf{S}_u$ are often interchanged (especially if I is large) and often without difference in notation: In either case the covariance estimate will be called $\mathbf{S}$.

B.1.3 Sample Correlation

The sample correlation coefficient between the ith and the jth variables is

$$r_{ij} = \frac{s_{ij}}{s_i s_j}. \tag{B.10}$$

This coefficient is—unlike s_{ij}—invariant under change of origin and scale of the variables.
The matrix

$$\mathbf{R} = (r_{ij}) \tag{B.11}$$

where $r_{ii} = 1$ and $|r_{ij}| \leq 1$, is called the sample correlation matrix. If the variables are uncorrelated we get $\mathbf{R} = \mathbf{I}$. When the variables are independent they are also uncorrelated, the converse is not true.

B.1.4 Scaling

If the data $\mathbf{X}$ is scaled by means of a linear transformation $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \mathbf{b}$, $\quad i = 1, \ldots, I$, which is equal to writing $\mathbf{Y} = \mathbf{X}\mathbf{A}^\top + \mathbf{1}\mathbf{b}^\top$. The mean and variance are given by

$$\bar{\mathbf{y}} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{b} \tag{B.12}$$

$$\mathbf{S}_y = \mathbf{A}\mathbf{S}\mathbf{A}^\top \tag{B.13}$$

One way of scaling data is to simply subtract the mean. This scaling is called *centring*. The caused the numercal values of the data entries to become smaller as they are positioned around 0 leading to higher accuracy of numerical algorithms. The constant term in regression model can also be eliminated when using centred data.

B.1.4.1 Autoscaling

When the variables in $\mathbf{x}_i$ have different units, the *units* of the covariances are mixed and can be difficult to interpret. The *sizes* of the elements in $\mathbf{S}$ depend on the choice of unit (e.g. inches or meters). If we let

$$\mathbf{y}_i = \mathbf{D}^{-1}(\mathbf{x}_i - \bar{\mathbf{x}}), \tag{B.14}$$

where

$$\mathbf{D} = \text{diag}(s_1, s_2, \ldots, s_J), \tag{B.15}$$

the variables $\mathbf{y}_i$ are scaled to have unit variance and are mean centered. The variables are said to be *autoscaled*. This eliminates the effect of the units of the variables $\mathbf{x}_r$ and we have $\mathbf{S}_y = \mathbf{R}$.

B.2 Distributions

B.2.1 Distribution Functions

The cumulative distribution function (CDF) associated with the random vector $\mathbf{x} = (x_1, \ldots, x_p)$ is the function F defined by

$$F(\mathbf{x}^\star) = \Pr(\mathbf{x} \leq \mathbf{x}^\star) = \Pr(x_1 \leq x_1^\star, \ldots, x_p \leq x_p^\star),$$

where Pr denotes *the probability of*.
A random vector $\mathbf{x}$ is absolutely continuous if a probability density function (PDF), $f(\mathbf{x})$, exists such that

$$F(\mathbf{x}) = \int_{-\infty}^{\mathbf{x}} f(\mathbf{u})\, d\mathbf{u}. \tag{B.16}$$

B.2.2 Expectation and Variance

If $\mathbf{x}$ is a random vector with PDF $f(\mathbf{x})$ then the expectation of a function $g(\mathbf{x})$ is

$$E\{g(\mathbf{x})\} = \int_{-\infty}^{\infty} g(\mathbf{x}) f(\mathbf{x})\, d\mathbf{x} \tag{B.17}$$

The vector $E(\mathbf{x}) = \boldsymbol{\mu}$ is the population mean vector of $\mathbf{x}$. The covariance between two random vectors can be defined as the matrix

$$C(\mathbf{x}, \mathbf{y}) = E\{(\mathbf{x} - \boldsymbol{\mu})(\mathbf{y} - \boldsymbol{\nu})^\top\}, \tag{B.18}$$

where $\boldsymbol{\mu} = E\{\mathbf{x}\}$ and $\boldsymbol{\nu} = E\{\mathbf{y}\}$. The population covariance matrix is

$$C(\mathbf{x}, \mathbf{x}) = V\{\mathbf{x}\} = \boldsymbol{\Sigma} = (\sigma_{ij}). \tag{B.19}$$

We write

$$\mathbf{x} \sim (\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{B.20}$$

to describe a random vector with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The population correlation matrix $\mathbf{P} = (\rho_{ij})$ is defined by the elements

$$\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}, \tag{B.21}$$

where $\rho_{ii} = 1$ and $|\rho_{ij}| \leq 1$.

B.2.3 The Multinormal Distribution

If we write the PDF of $N(\mu, \sigma^2)$, the univariate normal distribution, as $f(x) = \left(2\pi\sigma^2\right)^{-1/2} \exp(-\frac{1}{2}(x-\mu)\{\sigma^2\}^{-1}(x-\mu))$ then the extension to the multivariate case comes easily

$$f(\mathbf{x}) = |2\pi\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{B.22}$$

If a random vector has the PDF given in (B.22), it is said to have a multinormal (or multivariate normal) distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, and we write $\mathbf{x} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The choice of parameters comes from the fact that $E\{\mathbf{x}\} = \boldsymbol{\mu}$ and $V\{\mathbf{x}\} = \boldsymbol{\Sigma}$.

B.2.3.1 Central Limit Theorem

An infinite sequence of independent identically distributed random vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ from a distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$ will have a sample mean vector $\bar{\mathbf{x}}$ which is approximately normally distributed.

$$n^{-1/2} \sum_{r=1}^{n} (\mathbf{x}_r - \boldsymbol{\mu}) = n^{1/2}(\bar{\mathbf{x}} - \boldsymbol{\mu}) \xrightarrow{\mathrm{D}} N_p(\mathbf{0}, \boldsymbol{\Sigma}) \text{ for } n \to \infty, \tag{B.23}$$

where $\xrightarrow{\mathrm{D}}$ means convergence in distribution. We can also write

$$\bar{\mathbf{x}} \sim N_p(\boldsymbol{\mu}, \frac{1}{n}\boldsymbol{\Sigma}). \tag{B.24}$$

B.2.4 The Wishart Distribution

When we look at quadratic functions of the form $\mathbf{X}^\top\mathbf{C}\mathbf{X}$, where $\mathbf{X}$ is $(n \times p)$ and $\mathbf{C}$ is a symmetric matrix, the Wishart distribution appears.

When $\mathbf{C} = n^{-1}\mathbf{H}$ the quadratic form defines the sample covariance matrix $\mathbf{S}$ (the centring matrix $\mathbf{H}$ is defined in (B.7)).

If $\mathbf{M}$ can be written $\mathbf{M} = \mathbf{X}^\top\mathbf{X}$ where $\mathbf{X}$ comes from a $N_p(\mathbf{0}, \boldsymbol{\Sigma})$ distribution, then $\mathbf{M}$ has a Wishart distribution with scale matrix $\boldsymbol{\Sigma}$ and degrees of freedom parameter n.
The notation used is:

$$\mathbf{M} \sim W_p(\boldsymbol{\Sigma}, n). \tag{B.25}$$

If $\mathbf{M}$ is defined as $\mathbf{M} = \mathbf{X}^\top\mathbf{C}\mathbf{X}$ and $\mathbf{C}$ is an idempotent matrix then $\mathbf{M} \sim W_p(\mathbf{\Sigma}, r)$, where $r = \operatorname{tr}\mathbf{C} = \operatorname{rank}\mathbf{C}$.
The following relation holds for the sample covariance matrix: $n\mathbf{S} = W_p(\mathbf{\Sigma}, n-1)$.

B.2.5 The Hotelling T^2 Distribution

The Hotelling T^2 statistic is of the type $\mathbf{x}^\top\mathbf{M}^{-1}\mathbf{x}$, where $\mathbf{x}$ is normal, $\mathbf{M}$ is Wishart and $\mathbf{x}$ and $\mathbf{M}$ are independent.

If $\mathbf{x} \sim N_p(0, \mathbf{I})$ and $\mathbf{M} \sim W_p(\mathbf{I}, m)$, then $\alpha = m\mathbf{x}^\top\mathbf{M}^{-1}\mathbf{x}$ has a Hotelling T^2 distribution. The obvious notation is

$$\alpha \sim T^2(p, m) \tag{B.26}$$

A more general result can be found. If $\mathbf{x}$ and $\mathbf{M}$ are independently distributed as $N_p(\boldsymbol{\mu}, \mathbf{\Sigma})$ and $W_p(\mathbf{\Sigma}, m)$, respectively, then

$$m(\mathbf{x} - \boldsymbol{\mu})^\top\mathbf{M}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \sim T^2(p, m). \tag{B.27}$$

If $\bar{\mathbf{x}}$ and $\mathbf{S}$ are the mean vector and covariance matrix of a sample of size n from a $N_p(\boldsymbol{\mu}, \mathbf{\Sigma})$ distribution, then

$$\begin{aligned}(n-1)(\bar{\mathbf{x}} - \boldsymbol{\mu})^\top\mathbf{S}^{-1}(\bar{\mathbf{x}} - \boldsymbol{\mu}) &= \\ n(\bar{\mathbf{x}} - \boldsymbol{\mu})^\top\mathbf{S}_u^{-1}(\bar{\mathbf{x}} - \boldsymbol{\mu}) &\\ \sim T^2(p, n-1), &\end{aligned} \tag{B.28}$$

where $\mathbf{S}_u = n/(n-1)\,\mathbf{S}$.

The T^2 distribution is related to the F distribution by the following relation

$$T^2(p, m) = \frac{mp}{m - p + 1} F_{p, m-p+1} \tag{B.29}$$

B.3 Univariate Statistics

Many tests in the area of multivariate statistics can be boiled down to tests based on univariate distributions. The following sections list a few.

B.3.1 Chi-squared Distribution

Let $x_1, \ldots, x_p$ be independent $N(0, 1)$ variables and set $y = x_1^2 + \cdots + x_p^2$. Then y has a chi-squared distribution with p degrees of freedom.
This is written

$$y \sim \chi_p^2.$$

A stochastic variable has a Gamma distribution with parameters (a, b), $a > 0$, $b > 0$ if its PDF can be written as

$$f(x) = \frac{1}{G(a)a^b} x^{b-1} e^{-x/b} \qquad \text{for} \qquad x > 0 \tag{B.30}$$

and is equal to 0 otherwise. The function $G(a)$ is the Gamma function defined by

$$G(a) = \int_0^\infty t^{x-1} e^{-t}\, dt, \quad x > 0 \tag{B.31}$$

The χ^2 distribution is a special case of the Gamma distribution. A variable $y \sim \chi^2_p$ has the distribution $\Gamma(p/2, 2)$, where Γ denotes the Gamma distribution.

B.3.2 F and Beta Variables

Let $u \sim \chi^2_p$ and $v \sim \chi^2_q$ be independent variables. Set

$$x = \frac{u/p}{v/q}.$$

Then x is said to have a F distribution with degrees of freedom p and q.
We write

$$x \sim F_{p,q}.$$

If we set $y = u/(u+v)$ then y has a beta distribution with parameters $\frac{1}{2}p$ and $\frac{1}{2}q$.
In this case we write

$$y \sim B_{\frac{1}{2}p, \frac{1}{2}q}.$$

The beta distribution with parameters (a, b) has the PDF

$$f(x) = \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1}, \tag{B.32}$$

where $B(a, b)$ is the beta integral defined by

$$\int_0^1 t^{a-1}(1-t)^{b-1}\, dt. \tag{B.33}$$

The F and beta distributions are related by the transformation

$$y = \frac{px}{q+px} \tag{B.34}$$

B.3.3 t distribution

Let $x \sim N(0,1)$ independently of $u \sim \chi^2_p$. Then $t = \frac{x}{(u/p)^{1/2}}$ is said to have a t distribution with p degrees of freedom.
This is written

$$t \sim t_p.$$

The t distribution is related to the F distribution by the following relation: $t^2 \sim F_{1,p}$.

B.4 Bias/Variance dilemma

Data from a general process may be modelled as

$$y = g(\mathbf{x}) + \epsilon, \tag{B.35}$$

where $g(\mathbf{x})$ is some function (the *true* model) of the argument vector $\mathbf{x}$ defined by $g(\mathbf{x}) = E\{y|\mathbf{x}\}$, where E denotes the expectation. The conditional expectation $E\{y|\mathbf{x}\}$ defines the value of y that will be realised on average given a particular realisation of $\mathbf{x}$. ϵ is a random expectational error that represents our lack of knowledge about the dependence of y on $\mathbf{x}$. The only assumption on ϵ is that the expected value is zero.

The functional relationship of the *process model* is denoted $\mathcal{F}(\mathbf{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the parameters that needs to be estimated from the data obtained from the system.

A measure of the predictive capability of the process model is the squared distance between the true model $g(\mathbf{x})$ and the process model $\mathcal{F}(\mathbf{x}, \boldsymbol{\theta})$

$$(g(\mathbf{x}) - \mathcal{F}(\mathbf{x}, \boldsymbol{\theta}))^2 = (E\{y|\mathbf{x}\} - \mathcal{F}(\mathbf{x}, \boldsymbol{\theta}))^2.$$

The data set used for parameter estimation will be called D

$$D = \left[(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_I, y_I)\right].$$

It is noted that the parameters are a function of D and we will write $\mathcal{F}(\mathbf{x}, D)$ instead of $\mathcal{F}(\mathbf{x}, \boldsymbol{\theta})$.

We will now focus on the *mean square error* of the function $\mathcal{F}(\mathbf{x}, D)$ as an estimator for the function $g(\mathbf{x}) = E\{y|\mathbf{x}\}$, which is defined by

$$E_D\big\{(E\{y|\mathbf{x}\} - \mathcal{F}(\mathbf{x}, D))^2\big\}. \tag{B.36}$$

Here E_D represents the mean error over all the patterns in the calibration data set D. Using that the conditional expectation $E\{y|\mathbf{x}\}$ has constant expectation with respect to D we get

$$\begin{aligned}
&E_D\big\{(E\{y|\mathbf{x}\} - \mathcal{F}(\mathbf{x}, D))^2\big\} \\
&\quad = E_D\big\{\big(E\{y|\mathbf{x}\} - E_D\{\mathcal{F}(\mathbf{x}, D)\} + E_D\{\mathcal{F}(\mathbf{x}, D)\} - \mathcal{F}(\mathbf{x}, D)\big)^2\big\} \\
&\quad = E_D\big\{\big(E\{y|\mathbf{x}\} - E_D\{\mathcal{F}(\mathbf{x}, D)\}\big)^2\big\} + E_D\big\{\big(E_D\{\mathcal{F}(\mathbf{x}, D)\} - \mathcal{F}(\mathbf{x}, D)\big)^2\big\} \\
&\qquad + 2E_D\big\{\big(E\{y|\mathbf{x}\} - E_D\{\mathcal{F}(\mathbf{x}, D)\}\big)\big(E_D\{\mathcal{F}(\mathbf{x}, D)\} - \mathcal{F}(\mathbf{x}, D)\big)\big\} \\
&\quad = \big(E\{y|\mathbf{x}\} - E_D\{\mathcal{F}(\mathbf{x}, D)\}\big)^2 + E_D\big\{\big(\mathcal{F}(\mathbf{x}, D) - E_D\{\mathcal{F}(\mathbf{x}, D)\}\big)^2\big\} \\
&\qquad + 2\big(E\{y|\mathbf{x}\} - E_D\{\mathcal{F}(\mathbf{x}, D)\}\big)E_D\big\{E_D\{\mathcal{F}(\mathbf{x}, D)\} - \mathcal{F}(\mathbf{x}, D)\big\} \\
&\quad = \big(E_D\{\mathcal{F}(\mathbf{x}, D)\} - E\{y|\mathbf{x}\}\big)^2 + E_D\big\{(\mathcal{F}(\mathbf{x}, D) - E_D\{\mathcal{F}(\mathbf{x}, D)\})^2\big\}.
\end{aligned} \tag{B.37}$$

The cross term is cancelled by noting that

$$E_D\big\{E_D\{\mathcal{F}(\mathbf{x}, D)\} - \mathcal{F}(\mathbf{x}, D)\big\} = E_D\{\mathcal{F}(\mathbf{x}, D)\} - E_D\{\mathcal{F}(\mathbf{x}, D)\} = 0$$

The two terms in equation (B.37) account for the *squared bias* and *variance* of the approximation function $\mathcal{F}(\mathbf{x}, D)$ respectively. This is termed the bias/variance dilemma [Haykin, 1994].

If, on average, the approximation function $\mathcal{F}(\mathbf{x}, D)$ is different from the regression function $g(\mathbf{x})$. We call $\mathcal{F}(\mathbf{x}, D)$ a biased estimator of $g(\mathbf{x})$. If $E_D\{\mathcal{F}(\mathbf{x}, D)\} = g(\mathbf{x})$ the estimator is unbiased. The mean square error of an unbiased estimator may be large if the variance is large. A strategy is then to find a biased estimator that will have smaller variance such that the sum (the mean square error) will be smaller than when an unbiased estimator is used.

Introducing bias into an estimator is of course not a guarantee for decreasing the variance and/or the mean square error.

An application of regularisation to handle the bias/variance dilemma in parameter estimation for artificial neural networks can be found in [Sjöberg *et al.*, 1994; Sjöberg and Ljung, 1995].

References

Al-Salti, M. and Statham, A. (1994). A review of the literature on the use of SPC in batch production. *Quality and reliability engineering international*, **10**, 49–61.

Albert, S.; Martin, E.; Montague, A. and Morris, A. J. (1997). Multivariate statistical process control in batch process monitoring. In J. J. Gertler; J. B. Cruz, Jr.; M. Peshkin; M. Kümmel; R. Iserman; M. Perrier and A. Munack, editors, *Proceedings of the 13th World Congress, International Federation of Automatic Control.*, pages 389–394. Pergamon.

Alsberg, B. K.; Woodward, A. M. and Kell, D. B. (1997). An introduction to wavelet transforms for chemometricians: A time frequency approach. *Chemometrics and Intelligent Laboratory Systems*, **37**(2), 215–239.

Baffi, G.; Martin, E. B. and Morris, A. J. (1999a). Non-linear projection to latent structures revisieted (the neural network PLS algorithm). *Comp. Chem. Engng.*, **23**, 1293–1307.

Baffi, G.; Martin, E. B. and Morris, A. J. (1999b). Non-linear projection to latent structures revisited: the quadratic PLS algorithm. *Comp. Chem. Engng.*, **23**, 395–411.

Bakshi, B. R. and Stephanopoulos, G. (1994a). Representation of process trends—III. Multiscale extraction of trends from process data. *Comp. Chem. Engng.*, **18**(4), 267–302.

Bakshi, B. R. and Stephanopoulos, G. (1994b). Representation of process trends—IV. Induction of real-time patterns from operating data for diagnosis and supervisory control. *Comp. Chem. Engng.*, **18**(4), 303–322.

Bakshi, B. R.; Locher, G.; Stephanopoulos, G. and Stephanopoulos, G. (1994). Analysis of operating data for evaluation, diagnosis and control of batch operations. *J. Proc. Cont.*, **4**(4), 179–194.

Basilevsky, A. (1983). *Applied Matrix Algebra in the Statistical Sciences*. Elsevier Science Publishing Co.

Bennett, S. (1997). History of automatic control to 1960: an overview. In J. J. Gertler; J. B. Cruz; M. Peshkin; K. Furuta; K. H. Fasol; R. Bitmead and I. Petersen, editors, *Proceedings of the 13th World Congress, IFAC, Vol. G, Robust Control*, pages 117–122. Pergamon.

Berber, R. (1996). Control of batch reactors: A review. *Chemical Engineering Research and Design*, **74**(1), 3–20.

Björck, Å. (1994). Generalized and Sparse Least Squares Problems. In E. Spedicato, editor, *Algorithms for Continuous Optimization*, chapter 3, pages 37–80. Kluwer Academic Publishers.

Boqué, R. and Smilde, A. K. (1999). Monitoring and diagnosing batch processes with multiway covariates regerssion models. *AIChE Journal*, **45**(7), 1504–1520.

Box, G. E. P.; Jenkins, G. M. and Reinsel, G. C. (1994). *Time Series Analysis, Forecasting and Control*. Prentice Hall, third edition.

Bro, R. (1997). PARAFAC. Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, **38**(2), 149–172.

Chiang, L. H.; Russell, E. L. and Braatz, R. D. (2001). *Fault Detection and Diagnosis in Industrial Systems*. Springer, London.

Chin, I.; Lee, K. S. and Lee, J. H. (1998). A unified framework for control of batch processes. In *AIChE Annual Meeting, Miami*.

Çinar, A. and Undey, C. (1999). Statistical process and controller performance monitoring, A tutorial on current methods and future directions. In *Proceeding of the American Control Conference, San Diego, California, USA*, pages 2625–2639. IEEE.

Clarke, D. W. and Ghaoud, T. (2002). Validation of vortex flowmeters. *Computing and Control Engineering Journal*, **13**(5), 237–241.

Coxon, A. P. M. (1982). *A User's Guide to Multidimensional Scaling*. Heinemann Educational Books, Exeter.

de Veaux, R. D.; Psichogios, D. C. and Ungar, L. H. (1993). A comparison of two nonparametric estimation schemes: MARS and neural networks. *Comp. Chem. Engng.*, **17**(8), 819–837.

Dennis, J. E. and Schnabel, R. B. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice–Hall, Inc.

Dewilde, P. and van der Veen, A. (1998). *Time-Varying Systems and Computations*. Kluwer Academic Publishers, The Netherlands.

Dieci, L.; Morini, B. and Papini, A. (1996). Computational techniques for real logarithms of matrices. *SIAM J. Matrix Anal. Appl.*, **17**(3), 570–593.

Dong, D. and McAvoy, T. J. (1996). Nonlinear principal component analysis—based on principal curves and neural networks. *Comp. Chem. Engng.*, **20**(1), 65–78.

Duboc, P. (1997). *Transient Growth of Saccharomyces cerevisiae, a quantitative approach.* Ph.D. thesis, École Polytechnique Fédérale de Lausanne.

Fayyad, U.; Haussler, D. and Stolorz, P. (1996a). Mining scientific data. *Comm. of the ACM*, **39**(11), 51–57.

Fayyad, U.; Piatetsky-Shapiro, G. and Smyth, P. (1996b). The KDD Process for extracting useful knowledge from volumes of data. *Comm. of the ACM*, **39**(11), 27–34.

Frank, I. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, **35**(2), 109–135.

Geladi, P. and Kowalski, B. R. (1986). Partial least-squares regression: A tutorial. *Analytica Chimica Acta*, **185**, 1–17.

Glymour, C.; Madigan, D.; Pregibon, D. and Smyth, P. (1997). Statistical themes and lessons for data mining. *Data Mining and Knowledge Discovery*, **1**, 11–28.

Gollmer, K. and Posten, C. (1996). Supervision of bioprocesses using a dynamic time warping algorithm. *Control Engineering Practice*, **4**(9), 1287–1295.

Golub, G. H. and van Loan, C. F. (1991). *Matrix Computations.* The Johns Hopkins University Press.

Gregersen, L. and Jørgensen, S. B. (1999). Supervision of fed-batch fermentations. *The Chemical Engineering Journal*, **75**, 69–76.

Gregersen, L. and Jørgensen, S. B. (2002). Dynamic I/O Modelling for Batch Processes. *In preparation.*

Hansen, P. C. (1996). *Rank-Deficient and Discrete Ill-Posed Problems.* Polyteknisk Forlag, Lyngby, Denmark.

Hansen, P. C. (1998). Regularization Toolbox. Technical report, Department of Mathematical Modelling, Technical University of Denmark, `http://www.imm.dtu.dk/~pch`.

Haykin, S. (1994). *Neural Networks, A Comprehensive Foundation.* MacMillan College Publishing Company.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**(1), 55–67.

Höskuldsson, A. (1988). PLS regression methods. *Journal of Chemometrics*, **2**, 211–228.

Höskuldsson, A. (1994). The H-principle: new ideas, algorithms and methods in applied mathematics and statistics. *Chemom. Intell. Lab. Syst.*, **23**, 1–28.

Höskuldsson, A. (1995). A combined theory for PCA and PLS. *Journal of Chemometrics*, **9**, 91–123.

Höskuldsson, A. (1996). *Regression Methods in Science and Technology*, volume 1. Thor Publishing, Arnegaards Allé 7, Denmark.

Huo, Y.; Ioannou, P. A. and Mirmirani, M. (2001). Fault-Tolerant Control and Reconfiguration for High Performance Aircraft: Review. Technical report, Center for Advanced Transportation Technologies, University of Southern California, USA.

Ignova, M.; Paul, G. C.; Glassey, J.; Ward, A. C.; Montague, G. A.; Thomas, C. R. and Karim, M. N. (1996). Towards intelligent process supervision: Industrial penicillin fermentation case study. *Comp. Chem. Engng.*, **20, Suppl.**, S545–S550.

Jackson, J. E. (1991). *A User's Guide to Principal Components*. John Wiley & Sons, Inc.

Jørgensen, S. B. and Jensen, N. (1989). Dynamics and control of chemical reactors - selectively surveyed. In *Preprints DYCORD 89*, pages 359–371. IFAC.

Juricek, B. C.; Larimore, W. E. and Seborg, D. E. (1999). Reduced rank ARX and subspace system identification for process control. In C. Georgakis, editor, *IFAC Symposium on dynamics and control of process systems, DYCOPS-5*, pages 245–250. Elsevier Science.

Kourti, T. (2002). Process Analysis and Abnormal Situation Detection: From Theory to Practice. *IEEE Control Systems Magazine*, **22**(5), 10–25.

Kourti, T.; Nomikos, P. and MacGregor, J. F. (1995). Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS. *J. Proc. Cont.*, **5**(4), 277–284.

Kresta, J. V.; MacGregor, J. F. and Marlin, T. E. (1991). Multivariate statistical monitoring of process operating performance. *The Canadian Journal of Chemical Engineering*, **69**, 35–47.

Kroonenberg, P. M. (1983). *Three-Mode Principal Component Analysis*. DSWO Press.

Lakshminarayanan, S.; Shah, S. L. and Nandakumar, K. (1997). Modeling and control of multivariate processes: Dynamic PLS approach. *AIChE Journal*, **43**(9), 2307–2322.

Leahy, M. J.; Henry, M. P. and Clarke, D. W. (1997). Sensor validation in biomedical applications. *Control Engineering Practice*, **5**(12), 1753–1758.

Lee, K. S.; Lee, J. H.; Chin, I. and Lee, H. J. (1997). A model predictive control technique for batch processes and its application to temperature tracking control of an experimental batch reactor. In *AIChE Annual Meeting in Los Angeles*.

Lennox, B.; Hiden, H. G.; Montague, G. A.; Kornfeld, G. and Goulding, P. R. (2000). Application of multivariate statistical process control og batch operations. *Computers and Chemical Engineering*, **24**, 291–296.

Leung, A. K.; Chau, F.-T. and Gao, J.-B. (1998). A review on applications of wavelet transform techniques in chemical analysis: 1989–1997. *Chemometrics and Intelligent Laboratory Systems*, **43**(1–2), 165–184.

Little, R. J. A. and Rubin, D. B. (1987). *Statistical Analysis with Missing Data*. John Wiley & Sons, Inc.

Liu, K. (1997). Identification of linear time-varying systems. *Journal of Sound and Vibration*, **206**(4), 487–505.

Ljung, L. (1987). *System Identification, Theory for the User*. Prentice Hall.

Lorber, A. *et al.* (1987). A theoretical foundation for the PLS algorithm. *Journal of Chemometrics*, **1**, 19–31.

Louwerse, D. J. and Smilde, A. K. (2000). Multivariate statistical process control of batch processes based on three-way models. *Chemical Engineering Science*, **55**(7), 1225–1235.

Luo, R.; Misra, M. and Himmelblau, D. M. (1999). Sensor fault detection via multiscale analysis and dynamic PCA. *Ind. Eng. Chem. Res.*, **38**, 1489–1495.

MacGregor, J. F. (1997). Using on-line process data to improve quality: challenges for statisticians. *International Statistical Review*, **65**, 309–323.

MacGregor, J. F. and Kourti, T. (1995). Statistical process control of multivariate processes. *Control Eng. Practice*, **3**(3), 403–414.

MacGregor, J. F. and Nomikos, P. (1992). Monitoring batch processes. In *NATO ASI, Turkey, Batch Process Systems Engineering*.

MacGregor, J. F.; Jaeckle, C.; Kiparissides, C. and Koutoudi, M. (1994). Process monitoring and diagnosis by multiblock PLS methods. *AIChE Journal*, **40**(5), 826–838.

Mardia, K. V.; Kent, J. T. and Bibby, J. M. (1995). *Multivariate Analysis*. Academic Press Inc., second edition.

Martens, H. and Næs, T. (1989). *Multivariate Calibration*. John Wiley & Sons.

Martin, E.; Morris, J. and Lane, S. (2002). Monitoring Process Manufactoring Processes. *IEEE Control Systems Magazine*, **22**(5), 26–39.

Martin, E. B. and Morris, A. J. (1996). Non-parametric confidence bounds for process performance monitoring charts. *Journal of Process Control*, **6**(6), 349–358.

Martin, E. B.; Morris, A. J.; Papazoglou, M. C. and Kiparissides, C. (1996). Batch process monitoring for consistent production. *Comp. Chem. Engng.*, **20**, S599–S604. Suppl.

Mason, R. L.; Tracy, N. D. and Young, J. C. (1995). Decomposition of T^2 for multivariate control chart interpretation. *Journal of Quality Technology*, **27**(2), 99–108.

MATLAB (1999). *Using MATLAB*. The Mathworks Inc., 24 Prime Park Way Natick, MA, USA.

Miller, P.; Swanson, R. E. and Heckler, C. F. (1993). Contribution plots: A missing link in multivariate quality control. In *37th Annual Conference, ASCQ, NY*.

Moler, C. and van Loan, C. (1978). Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, **20**(4), 801–836.

Mulholland, M.; Hibbert, D. B.; Haddad, P. R. and Sammut, C. (1995). Application of the C4.5 classifier to building an expert system for ion chromotography. *Chemometrics and Intelligent Laboratory Systems*, **27**, 95–104.

Negiz, A. and Çinar, A. (1998). Monitoring of multivariate dynamic processes and sensor auditing. *Journal of Process Control*, **8**(5–6), 375–380.

Neogi, D. and Schlags, C. E. (1997). Application of mulivariate statistical techniques for monitoring emulsion batch processes. In *Proceedings of the American Control Conference, Albuquerque, New Mexico*, pages 1177–1181. IEEE.

Nomikos, P. (1995). *Statistical Process Control of Batch Processes*. Ph.D. thesis, McMaster University.

Nomikos, P. and MacGregor, J. F. (1995). Multivariate SPC charts for monitoring batch processes. *Technometrics*, **37**(1), 41–59.

Orr, K. (1998). Data quality and systems. *Comm. of the ACM*, **41**(2), 66–71.

Pond, R. J. (1994). *Fundamentals of Statistical Quality Control*. MacMillan College Publishing Company.

Press, W. H.; Teukolsky, S. A.; Vetterling, W. T. and Flannery, B. P. (1992). *Numerical Recipes in C*. Cambridge University Press, second edition.

Pyle, D. (1999). *Data Preparation for Data Mining*. Morgan Kaufmann Publishers.

Ramsay, J. O. and Silverman, B. W. (1997). *Functional Data Analysis*. Springer Series in Statistics. Springer Verlag.

Rani, K. Y. and Rao, V. S. R. (1999). Control of fermenters - a review. *Bioprocess Engineering*, **21**, 77–88.

Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. John Wiley & Sons.

Rugh, W. J. (1996). *Linear System Theory*. Prentice-Hall, Inc.

Rusell, S. A.; Robertson, D. G.; Lee, J. H. and Ogunnaike, B. A. (2000). Model-based quality monitoring of batch and semi-batch processes. *Journal of Process Control*, **10**, 317–332.

Russell, S. A.; Kesavan, P.; Lee, J. H. and Ogunnaike, B. A. (1998). Recursive data-based prediction and control of batch product quality. *AIChE Journal*, **44**(11), 2442–2458.

Saner, U. and Stephanopoulos, G. (1992). Application of Pattern Recognition Techniques to Fermentation Data Analysis. In *IFAC Modelling and Control of Biotechnical Processes, Colorado, USA*, pages 123–128.

Schaper, C. D.; Larimore, W. E.; Seborg, D. E. and Mellichamp, D. A. (1990). Identification of chemical processes using canonical variate analysis. In *Proceedings of the 29th Conference on Decision and Control*, pages 605–610, Honolulu, Hawaii.

Shao, R.; Jia, F.; Martin, E. B. and Morris, A. J. (1999). Wavelets and non-linear pricipal components analysis for process monitoring. *Control Engineering Practice*, **7**, 865–879.

Shimizu, K. (1993). An overview on the control system design of bioreactors. *Advances in Biochemical Engineering Biotechnology*, **50**, 65–84.

Simoglou, A.; Martin, E. B. and Morris, A. J. (1999). A comparison of canonical variate analysis and partial least squares for the identification of dynamic processes. In *Proceedings of the American Control Conference*, pages 832–837, San Diego, California, USA.

Simoglou, A.; Martin, E. B. and Morris, A. J. (2002). Statistical performance monitoring of dynamic multivariate processes using state space modelling. *Computers and Chemical Engineering*, **26**, 909–920.

Simonoff, J. S. (1996). *Smoothing Methods in Statistics*. Springer.

Sjöberg, J. and Ljung, L. (1995). Overtraining, regularization and searching for a minimum, with application to neural networks. *Int. J. Control*, **62**(6), 1391–1407.

Sjöberg, J.; Hjalmarsson, H. and Ljung, L. (1994). Neural Networks in System Identification. In M. Blanke and T. Söderström, editors, *SYSID '94, 10th IFAC Symposium on System Identification*, pages 49–72.

Soroush, M. (1998). State and parameter estimations and their applications in process control. *Comp. Chem. Engng.*, **23**, 229–245.

Svozi, D.; Kvasnicka, V. and Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Labroratory Systems*, **39**(1), 43–62.

Tates, A. A.; Louwerse, D. J.; Smilde, A. K.; Koot, G. L. M. and Berndt, H. (1999). Monitoring a PVC batch process with mutivariate statistical process control charts. *Ind. Eng. Chem. Res.*, **38**, 4769–4776.

Tracy, N. D. and Young, J. C. (1992). Multivariate control charts for individual observations. *Journal of Quality Technology*, **24**(2), 88–95.

Tufte, E. R. (2001). *The Visual Display of Quantitative Information*. Graphics Press, Chesire, Connecticut, USA.

Unbehauen, H. and Rao, G. P. (1998). A review of identification in continuous-time systems. *Annual Reviews in Control*, **22**, 145–171.

Undey, C. and Cinar, A. (2002). Statistical Monitoring of Multistage, Multi-purpose Batch Processes. *IEEE Control Systems Magazine*, **22**(5), 40–52.

van Loan, C. F. (1994). Computing Integrals Involving the Matrix Exponential. In R. V. Patel; A. J. Laub and P. M. van Dooren, editors, *Numerical Linear Algebra Techniques for Systems and Control*, pages 681–690, New York. IEEE Press.

Verhaegen, M. and Yu, X. (1995). A class of subspace model identification algorithms to identify periodically and arbitrarily time-varying systems. *Automatica*, **31**(2), 201–216.

Wang, X. Z. and McGreavy, C. (1998). Automatic Classification for Mining Process Operational Data. *Industrial and Engineering Chemistry Research*, **37**(6), 2215–2222.

Watson, M. J.; Liakopoulos, A.; Brzakovic, D. and Georgakis, C. (1998). A Practical Assessment of Process Data Compression Techniques. *Industrial & Engineering Chemistry Research*, **37**(1), 267–274.

Westerhuis, J. A.; Kourti, T. and MacGregor, J. F. (1999). Comparing alternate approaches for multivariate statistical analysis of batch process data. *Journal of Chemometrics*, **13**, 397–413.

Wise, B. M. (1991). *Adapting multivariate analysis for monitoring and modelling of dynamic systems*. Ph.D. thesis, University of Washington, Seattle, USA.

Wise, B. M. and Gallagher, N. B. (1996). The process chemometrics approach to process monitoring and fault detection. *Journal of Process Control*, **6**(6), 329–348.

Wold, H. (1966). Nonlinear Estimation by Iterative Least Square Procedures. In F. N. David, editor, *Research Papers in Statistics*, pages 411–444. John Wiley & Sons.

Wold, S. (1993). Discussion: PLS in chemical practice. *Technometrics*, **35**(2), 136–139.

Wold, S. (1995). Chemometrics; what do we mean with it, and what do we want from it. *Chemometrics and Intelligent Laboratory Systems*, **30**, 109–115.

Wold, S. *et al.* (1987a). Multi-way principal components and PLS-analysis. *Journal of Chemometrics*, **1**, 41–56.

Wold, S. *et al.* (1989). Nonlinear PLS modeling. *Chemom. Intell. Lab. Syst.*, **7**, 53–65.

Wold, S.; Esbensen, K. and Geladi, P. (1987b). Principal component analysis. *Chem. Intell. Lab. Syst.*, **2**, 37–52.

Wold, S.; Sjostrom, M. and Eriksson, L. (2001). PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, **58**, 109–130.

Zorriassantine, F. and Tannock, J. D. T. (1998). A review of neural networks for statistical process control. *Journal of Intelligent Manufactoring*, **9**, 209–224.